



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Problema de rutas de vehículos

Autor/es

ERNESTO PÉREZ SANCHO

Director/es

JESÚS MARÍA ARANSAY AZOFRA

Facultad

Facultad de Ciencia y Tecnología

Titulación

Grado en Matemáticas

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2018-19



Problema de rutas de vehículos, de ERNESTO PÉREZ SANCHO
(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative
Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.
Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los
titulares del copyright.



UNIVERSIDAD DE LA RIOJA

Facultad de Ciencia y Tecnología

TRABAJO FIN DE GRADO

Grado en Matemáticas

Problema de rutas de vehículos

Realizado por:

Ernesto Pérez Sancho

Tutelado por:

Jesús María Aransay Azofra

Logroño, julio de 2019

Resumen

A lo largo de este trabajo fin de grado nos vamos a centrar en presentar tres métodos de solución del Problema de rutas de vehículos con capacidad (CVRP), el cual consiste en encontrar la ruta óptima para una serie de vehículos con capacidad de carga limitada que deben visitar un número de puntos cada uno con una demanda concreta.

En primer lugar estudiaremos el artículo seminal que escribieron Dantzig y Ramser en 1959 en el cual se formula el problema y se da un algoritmo aproximado de solución del mismo. Nosotros vamos a centrarnos en el algoritmo presentado viendo con detalle cómo los autores lo explican sobre un caso de estudio concreto.

Después presentaremos dos métodos adicionales de solución, de los conocidos como heurísticos, basados en aportar una solución inicial sobre la cual se van haciendo mejoras iterativamente por medio de una función de ahorro. Los métodos presentados son los enunciados por Clarke y Wright (1964) y Altinkemer y Gavish (1991).

Abstract

Throughout this final degree project we will focus on presenting three methods of solving the Capacitated Vehicle Routing Problem (CVRP), which is to find the optimal route for a series of vehicles with limited load capacity. Each one must visit a number of points with a specific demand.

In the first place we will study the seminal article written by Dantzig and Ramser in 1959 in which the problem is formulated and an approximate algorithm for solving it is given. We are going to focus on the algorithm presented by looking in detail at how the authors explain it about a specific case study.

Then we present two additional methods of solution, known as heuristics, based on providing an initial solution on which improvements are made iteratively by means of a saving function. The methods presented are those enunciated by Clarke and Wright (1964) and Altinkemer and Gavish (1991).

Índice general

Introducción y motivación	1
Justificación económica y medioambiental	3
Complejidad computacional	3
Variantes y métodos de solución	5
Organización de la memoria	7
 1. Algoritmo de Dantzig y Ramser	 9
1.1. Conceptos básicos	9
1.2. Presentación del problema	10
1.3. Algoritmo	13
1.4. Problemas relacionados	26
 2. Algoritmo de Clarke y Wright	 27
2.1. Conceptos básicos	28
2.2. Presentación del problema	28
2.3. Procedimiento computacional	31
2.4. Mejoras del método de solución	35
 3. Heurística de ahorro paralelo	 37
3.1. El problema de la entrega	37
3.2. Heurística de ahorro paralelo	39
3.3. Ampliación: uso de nodos fantasma	46
 Conclusiones y Trabajo Futuro	 49
 A. Ejemplo numérico de Dantzig y Ramser	 55
A.1. Primera etapa de agregaciones	55
A.2. Segunda etapa de agregaciones	61
 B. Ejemplo numérico de Clarke y Wright	 81
 C. Problema de emparejamiento de peso máximo	 89
C.1. Emparejamientos en grafos	89
C.2. Algoritmos	89

Índice de figuras

1.	W.R. Hamilton.	1
2.	T.P. Kirkman.	1
3.	K. Menger.	1
4.	G. Dantzig (1914-2005).	2
5.	Almacén con pasillos cruzados.	4
6.	Forma de S.	4
7.	Punto medio.	5
8.	Ejemplo de VRPPD.	6
1.1.	Ejemplo de un grafo.	9
1.2.	Tabla del ejemplo al inicio.	15
1.3.	Corrección rápida geoméricamente.	16
1.4.	Tabla del ejemplo tras las correcciones rápidas.	17
1.5.	Tabla de los correspondientes valores de la funciones Delta al inicio de la primera etapa de agregaciones	19
1.6.	Tabla con la primera corrección debido a la variable θ	20
1.7.	Tabla con la primera asignación de los valores de θ	21
1.8.	Tabla al final de la primera etapa de agregaciones.	21
1.9.	Tabla de los correspondientes valores de la funciones Delta al final de la primera etapa de agregaciones.	22
1.10.	Resultado de la primera fase de agregaciones.	22
1.11.	Tabla de ejemplo al inicio de la segunda etapa de agregaciones.	23
1.12.	Tabla de ejemplo durante la segunda etapa de agregaciones.	24
1.13.	Diagrama con las relaciones.	25
1.14.	Tabla durante el tratamiento fraccionarios de los $x_{i,j}$	25
1.15.	Tabla de ejemplo tras el primer tratamiento fraccionario de $x_{i,j}$	25
2.1.	Posiciones factibles.	29
2.2.	1ª descomposición.	29
2.3.	2ª descomposición.	29
2.4.	3ª descomposición.	30
2.5.	4ª descomposición.	30
2.6.	Tablas del ejemplo correspondientes al principio del algoritmo.	31
2.7.	Tablas en la segunda iteración.	33
2.8.	Tablas en la tercera iteración.	34
2.9.	Tablas al final del algoritmo.	35
3.1.	Matriz de distancias al inicio del ejemplo.	45
3.2.	Representación gráfica de los puntos al inicio del ejemplo.	45

3.3. Matriz de costos tras la primera iteración	45
3.4. Representación gráfica de los puntos tras la primera iteración. . .	46
3.5. Matriz de costos tras la segunda iteración	46
3.6. Representación gráfica de los puntos tras la segunda iteración . .	47
3.7. Tabla de soluciones para diferentes valores de T	48
A.1. Tabla del ejemplo al inicio.	55
A.2. Tabla tras la primera corrección.	56
A.3. Tabla tras la segunda corrección.	56
A.4. Tabla tras la tercera corrección.	57
A.5. Tabla tras la cuarta corrección.	57
A.6. Tabla tras el primer cálculo de la función Delta.	58
A.7. Tabla al introducir la primera variable θ	58
A.8. Tabla tras darle valor a la primera variable θ	59
A.9. Tabla tras el segundo cálculo de la función Delta.	59
A.10. Tabla al intentar introducir la segunda variable θ	59
A.11. Tabla tras intentar darle valor a la segunda variable θ	60
A.12. Tabla al introducir la segunda variable θ	60
A.13. Tabla tras darle valor a la segunda variable θ	60
A.14. Tabla tras el tercer cálculo de la función Delta.	62
A.15. Tabla al introducir la tercera variable θ	62
A.16. Tabla tras darle valor a la tercera variable θ	62
A.17. Tabla tras el cuarto cálculo de la función Delta.	63
A.18. Tabla al introducir la cuarta variable θ	63
A.19. Tabla tras darle valor a la cuarta variable θ	63
A.20. Tabla tras el quinto cálculo de la función Delta.	64
A.21. Tabla al introducir la quinta variable θ	64
A.22. Tabla tras darle valor a la quinta variable θ	64
A.23. Tabla tras el sexto cálculo de la función Delta.	65
A.24. Tabla al introducir la sexta variable θ	65
A.25. Tabla tras darle valor a la sexta variable θ	65
A.26. Tabla tras el séptimo cálculo de la función Delta.	66
A.27. Tabla al introducir la séptima variable θ	66
A.28. Tabla tras darle valor a la séptima variable θ	66
A.29. Tabla tras el último cálculo de la función Delta.	67
A.30. Tabla al intentar introducir la última variable θ	67
A.31. Tabla al inicio de la segunda etapa de agregaciones.	67
A.32. Tabla durante la segunda etapa de agregaciones.	69
A.33. Tabla durante la segunda etapa de agregaciones.	69
A.34. Tabla durante la segunda etapa de agregaciones.	69
A.35. Tabla durante la segunda etapa de agregaciones.	70
A.36. Tabla durante la segunda etapa de agregaciones.	70
A.37. Tabla durante la segunda etapa de agregaciones.	70
A.38. Tabla durante la segunda etapa de agregaciones.	70
A.39. Tabla durante la segunda etapa de agregaciones.	70
A.40. Tabla durante la segunda etapa de agregaciones.	71
A.41. Tabla durante la segunda etapa de agregaciones.	71
A.42. Tabla durante la segunda etapa de agregaciones.	71
A.43. Tabla durante la segunda etapa de agregaciones.	71
A.44. Tabla durante la segunda etapa de agregaciones.	71

A.45.Tabla durante la segunda etapa de agregaciones.	72
A.46.Tabla durante la segunda etapa de agregaciones.	72
A.47.Tabla durante la segunda etapa de agregaciones.	72
A.48.Tabla durante la segunda etapa de agregaciones.	72
A.49.Tabla durante la segunda etapa de agregaciones.	72
A.50.Tabla durante la segunda etapa de agregaciones.	73
A.51.Tabla durante la segunda etapa de agregaciones.	73
A.52.Tabla durante la segunda etapa de agregaciones.	73
A.53.Tabla durante la segunda etapa de agregaciones.	73
A.54.Tabla durante la segunda etapa de agregaciones.	73
A.55.Tabla durante la segunda etapa de agregaciones.	74
A.56.Tabla durante la segunda etapa de agregaciones.	74
A.57.Tabla durante la segunda etapa de agregaciones.	74
A.58.Tabla durante la segunda etapa de agregaciones.	74
A.59.Tabla durante la segunda etapa de agregaciones.	74
A.60.Tabla durante la segunda etapa de agregaciones.	75
A.61.Tabla durante la segunda etapa de agregaciones.	75
A.62.Tabla durante la segunda etapa de agregaciones.	75
A.63.Tabla durante la segunda etapa de agregaciones.	75
A.64.Tabla durante la segunda etapa de agregaciones.	75
A.65.Tabla durante la segunda etapa de agregaciones.	76
A.66.Tabla durante la segunda etapa de agregaciones.	76
A.67.Tabla durante la segunda etapa de agregaciones.	76
A.68.Diagrama con las relaciones.	77
A.69.Tabla durante la segunda etapa de agregaciones.	78
A.70.Tabla durante la segunda etapa de agregaciones.	78
A.71.Tabla durante la segunda etapa de agregaciones.	78
A.72.Tabla durante la segunda etapa de agregaciones.	78
A.73.Tabla durante la segunda etapa de agregaciones.	78
A.75.Diagrama con las relaciones.	79
A.74.Diagrama con las relaciones.	79
B.1. Tablas al principio del algoritmo.	82
B.2. Tablas en la segunda iteración.	82
B.3. Tablas en la tercera iteración.	83
B.4. Tablas en la cuarta iteración.	84
B.5. Tablas en la quinta iteración.	85
B.6. Tablas en la sexta iteración.	86
B.7. Tablas en la séptima iteración.	87
B.8. Tablas en la octava iteración.	87
B.9. Tablas en la última iteración.	88
C.1. Ejemplo de emparejamientos.	90

Introducción y motivación

El problema del vendedor viajero o TSP (por sus siglas en inglés, “Traveling Salesman Problem”) es uno de los problemas más importantes dentro de la optimización combinatoria para el cálculo de rutas. La idea principal de este problema es responder a la pregunta:

“Dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible para visitar cada ciudad exactamente una vez y al finalizar regresar a la ciudad de origen?”

Su origen no está claro, se piensa que fue definido por primera vez a principios del siglo XIX por el matemático irlandés W.R. Hamilton (ver Figura 1) y por el matemático británico T.P. Kirkman (ver Figura 2), ya que el desafío conocido como “icosian game” es un juego desarrollado por el primero de estos en 1857 y su idea es algo similar al TSP. Sin embargo, todo hace indicar que la forma general del TSP fue estudiada por primera vez por una serie de matemáticos en Viena y Harvard alrededor del año 1930, entre los que destaca el austriaco K. Menger (ver Figura 3).

En este trabajo vamos a tratar una generalización de este problema denominado el problema de enrutamiento de vehículos o VRP (por sus siglas en inglés, Vehicle Routing Problem). Este es un problema de optimización combinatoria y programación entera que contesta a la pregunta:



Figura 1: W.R. Hamilton.



Figura 2: T.P. Kirkman.



Figura 3: K. Menger.

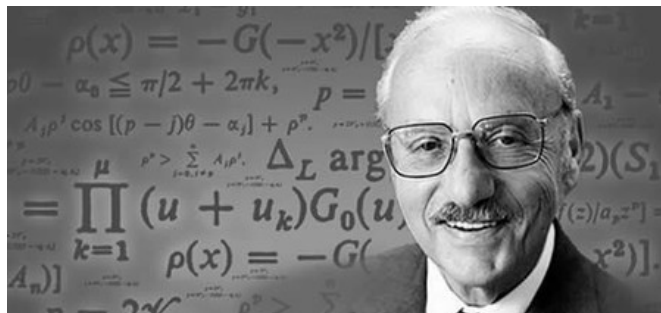


Figura 4: G. Dantzig (1914-2005).

“Dado un conjunto de vehículos y de clientes que solicitan una serie de demandas en unos puntos dados, ¿cuál es el conjunto óptimo de rutas para que dichos vehículos recorran esos puntos satisfaciendo las demandas solicitadas?”

La primera definición de este problema apareció en el trabajo de G. Dantzig (ver Figura 4) y J. Ramser [6] en 1959 donde plantean una aproximación algorítmica aplicada a la entrega de gasolina, el cual vamos a exponer posteriormente. Como curiosidad cabe destacar que años más tarde, en 1964, Clarke y Wright mejoraron dicha aproximación con un algoritmo al que se le denominó algoritmo de ahorros dando origen a lo que se conoce actualmente como “algoritmos heurísticos”. Con el paso del tiempo se han ido definiendo múltiples variantes del problema original (con entregas en ventanas de tiempo, con varios puntos de origen y destino...), algunos de los cuales enunciaremos en esta Introducción (ver Sección Variantes).

Para ver la importancia de este problema, lo primero que tenemos que tener en cuenta es que la población mundial es de casi 7 mil millones de personas y se prevé que alcance los 9 mil millones para 2040. Una característica importante de la población de clase media es que hay muchas personas y hay una gran demanda de recursos. Según las previsiones, este segmento de la población aumentará en 3 mil millones en los próximos 20 años, lo que aumentará el uso de los recursos solicitados de manera exponencial. En 2012, el Panel sobre Sostenibilidad Global de la Organización de las Naciones Unidas (ONU) llamó la atención sobre un hecho importante: el mundo requiere asegurar sus necesidades globales y estas necesidades son esencialmente el agua, los alimentos y la energía. Sin embargo, hay un aspecto de gran relevancia que no se tiene en cuenta y que está relacionado con la búsqueda continua de bienestar social: la movilidad. La poca infraestructura vial planificada y los vehículos no controlados que incorporan esta infraestructura, lleva a la humanidad a un estancamiento global. Hasta la fecha cuenta con casi 2 mil millones de vehículos, según las proyecciones de Ford, en 2040 serán 4 mil millones. Así que vemos que este problema destaca por diferentes motivos, entre estos las principales justificaciones son la económica y medioambiental, la complejidad computacional y las diferentes variantes que se pueden aplicar. A continuación vamos a ampliarlas.

Justificación económica y medioambiental

En muchas organizaciones, la gestión de las actividades de recogida y/o distribución constituye un importante problema de toma de decisiones. Con el paso de los años, las empresas se han dado cuenta de la repercusión de dicha actividad en sus costes totales, especialmente con la subida del petróleo. Todas las empresas se han preguntado alguna vez: ¿cuántos vehículos necesito para cubrir la demanda de mis clientes con el menor coste posible? o, ¿en qué orden debo visitar a éstos de forma que se minimicen mis costes? La gestión de rutas de la mayoría de las empresas se determina basándose en la experiencia del encargado de rutas, quien de manera intuitiva suele decidir cómo llevar a cabo la recogida y/o distribución, lo cual puede generar una gran pérdida económica. La informatización, automatización y optimización de dicha tarea implicaría, sin duda, una mejora en cuanto a la propia determinación y gestión de las rutas, así como una ventaja competitiva por disminución de costes.

Por ejemplo, los resultados del Estudio Nacional de Tendencias Logísticas en Colombia [20], Miebach Consulting, revelan que en Colombia el costo logístico respecto a las ventas oscila entre el 5 % y 10 %. Por otro lado, realizaron el mismo estudio en Centro América y en el Caribe y nos dicen que este indicador está aproximadamente entre el 3 % y el 5 %.

Para lograr disminuir este porcentaje se requiere idear un modelo que permita identificar una ruta eficiente de distribución de productos para todos los vehículos que se dispongan y para un número conocido de clientes. De esta manera, en la parte concreta de distribución, se podría lograr una reducción en la proporción de los costos logísticos sobre las ventas, un aumento en el nivel de servicio actual y con esto, disminuir los costos de los pedidos no entregados, al igual que una disminución en los tiempos de entrega. En especial, P. Toth y D. Vigo [24] concluyen que el uso de métodos computarizados en los procesos de distribución reales suelen disminuir los costos de transporte entre un 5 % y un 20 % y describen varios casos de estudio reales donde la aplicación de algoritmos VRP logra disminuciones sustanciales en los costos de transporte.

Además de todo lo contado, otra justificación económica está relacionada con una justificación medioambiental, ya que el transporte es uno de los aspectos más visibles dentro de la cadena de suministro. La cantidad de emisiones de dióxido de carbono (CO_2) del transporte se calcula en un 14 % de las emisiones totales. El transporte también es la principal fuente de emisión de dióxido de nitrógeno (NO_2), dióxido de azufre (SO_2) y otras partículas.

Complejidad computacional

J.A. Tompkins [14] cuenta que el proceso de planificación de los pedidos es el más laborioso de todos los procesos de almacén, ya que el tiempo de viaje ocupa en promedio el 50 % del tiempo total (tiempo que pasa desde que se realiza un pedido hasta que se entrega) y es recomendable minimizarlo.

Por ejemplo, si consideramos un ejemplo similar al VRP tomando una ca-

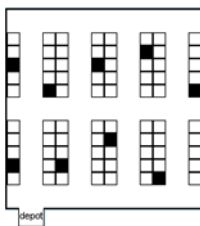


Figura 5: Almacén con pasillos cruzados.

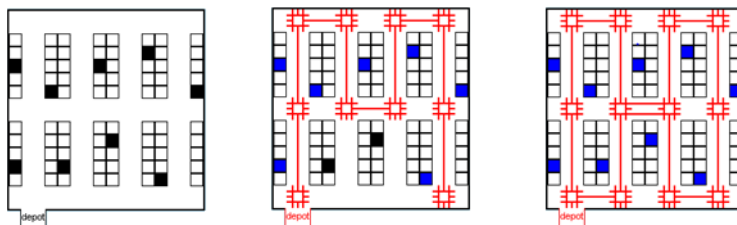


Figura 6: Forma de S.

retilla elevadora como vehículo y a los clientes como los lugares a donde tiene que ir a recoger una serie de artículos, teniendo que volver al punto de inicio, vemos que resultaría muy difícil encontrar rutas óptimas manualmente (ver Figura 5, 6 y 7). Cuantos más lugares hay para visitar, hay más permutaciones (es decir, hay más posibles rutas para pasar por los mismos puntos en distinto orden), tantas que en algunos casos puede ser muy complejo, incluso para algoritmos informáticos, generar un enrutamiento óptimo. Para ilustrar este ejemplo se puede aplicar a un almacén con pasillos cruzados en los que se incluyen una serie de artículos a recoger que se representan con un cuadrado en negro (ver Figura 5).

Según K.J. Roodbergen [19], profesor de logística cuantitativa en la Universidad de Groningen, existen diferentes métodos para su resolución. Algunos de estos métodos son:

- **Forma de S:** Consiste en recorrer todos los pasillos en los que aparecen entregas por la parte superior en forma de S (rodeándolos completamente), como se ve en la Figura 6. Posteriormente se hace lo mismo para los pasillos inferiores, hasta llegar al punto de inicio.
- **Punto medio:** Consiste en recorrer todos los pasillos empezando por la parte superior, recorriendo desde el punto más alto hasta donde aparece la entrega y luego volviendo al mismo punto, como se ve en la Figura 7. Posteriormente se hace lo mismo en la parte inferior de forma análoga pero empezando desde el punto más bajo.

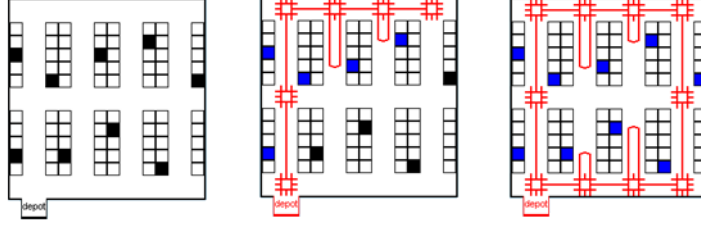


Figura 7: Punto medio.

Como conclusión, K.J. Roodbergen [19] dice que si por ejemplo utilizamos este método de enrutamiento con 7 pasillos, la longitud del pasillo de 30 metros y solo 10 elementos para elegir, la diferencia porcentual entre el método de enrutamiento manual y la ruta óptima tiene un error en promedio del 13 %. Por lo que vemos que en el caso de que nos queramos guiar por nuestra intuición tendremos una eficiencia baja, la cual disminuye a medida que se aumenta los diferentes factores que participan.

Cabe destacar que el VRP es conocido por ser NP-duro [15]. Para aclarar un poco el concepto de NP-duro se puede decir que si podemos encontrar un algoritmo A que resuelve uno de los problemas H de tipo NP-duro en tiempo polinómico (en computación, cuando el tiempo de ejecución de un algoritmo es menor que un cierto valor calculado a partir del número de variables implicadas usando una fórmula polinómica), entonces es posible construir un algoritmo que trabaje en tiempo polinómico para cualquier problema de tipo NP ejecutando primero la reducción de este problema en H y luego ejecutando el algoritmo A .

Sobre este aspecto también inciden Dantzig y Ramser [6] en su trabajo seminal sobre el CVRP, donde comentan que, asumiendo que en una ruta cada par de puntos está unido por un enlace, el número total de rutas diferentes a través de n puntos es $\frac{1}{2}n!$. Esto se puede demostrar contando el número de inicios y de finales de cada ruta, teniendo en cuenta que la ruta que va de un punto A hasta un punto B (por ejemplo A-D-C-B) es la misma que la misma ruta inversa (B-C-D-A), entonces eso es el número combinatorio $\binom{n}{2}$. Así ya dejamos de contar dos vértices, por lo que tenemos que multiplicar por $(n-2)!$, que es el número de vértices que nos quedan por medio donde sí nos importa el orden, por lo que nos queda lo dicho anteriormente:

$$\binom{n}{2} \cdot (n-2)! = \frac{n! \cdot (n-2)!}{(n-2)! \cdot 2!} = \frac{1}{2}n!.$$

Así que vemos que hasta para pequeños valores de n el número total de rutas es extremadamente grande, por ejemplo para $n = 10$, hay 1814400 rutas diferentes.

Variantes y métodos de solución

Todos los algoritmos y trabajos que presentamos en esta memoria hacen referencia al **Problema de Enrutamiento de Vehículos con Capacidad**

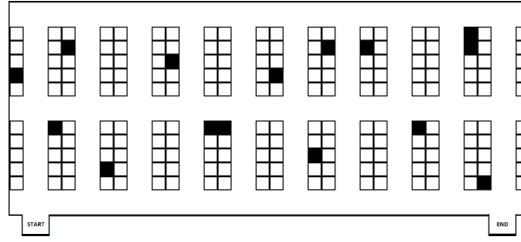


Figura 8: Ejemplo de VRPPD.

(**CVRP, Capacitated Vehicle Routing Problem**), que es una variante del VRP, en el cual los vehículos limitan su capacidad respecto a los bienes que tienen que ser entregados, y cada nodo tiene también demanda concreta. Este es el problema que plantean Dantzig y Ramser [6], Clarke y Wrighth [5] y Altinkemer y Gavish [18]. Otras variantes conocidas y estudiadas en la literatura son:

- **Problema de Enrutamiento del Vehículo con Recogida y Entrega (VRPPD, Vehicle Routing Problem with Pickup and Delivery):** Un número de productos necesita ser movido de cierta ubicación de recogida hacia otras ubicaciones de entrega. El objetivo es encontrar las rutas óptimas para un conjunto de vehículos para visitar las ubicaciones de recogida y entregar los productos en sus correspondientes ubicaciones (ver Figura 8). P.P. Ballesteros y A. Escobar [27] muestran una ampliación de este problema, la cual nos dice que fue tratado por primera vez por H. Min [8] en el año 1989, donde se reconoce la posibilidad de entregas y recogidas simultáneas en el mismo nodo. Este tiene una variante que surge de colocar una restricción adicional, que es la carga de los vehículos (en cualquier lugar de entrega, el artículo que se entrega debe ser el artículo que se recogió más recientemente) y que recibe el nombre de **Problema de enrutamiento del vehículo con LIFO**. Este problema reduce los tiempos de carga y descarga en las ubicaciones de entrega porque no hay necesidad de descargar temporalmente otros elementos que no sean los que se deben dejar. Benavent y otros [3] hicieron una explicación detallada de este problema.
- **Problema de Enrutamiento del Vehículo con Ventanas de Tiempo (VRPTW, Vehicle Routing Problem with Time Windows):** Las ubicaciones de entrega tienen ventanas de tiempo, dentro de las cuales las entregas tiene que ser realizadas. La primera formulación de este problema se remonta al año 1992 por Desrochers, Desrosiers y Solomon [21].
- **Problema de Enrutamiento de Vehículo Abierto (OVRP, Open Vehicle Routing Problem):** Las rutas de los vehículos no necesitan terminar en el depósito, pudiendo acabar en cualquier otro punto. Este problema fue formulado por primera vez a partir de dos documentos escritos por Fu y otros [30] y Li y otros [4].
- **Problema de Enrutamiento del Vehículo con Periodicidad (PVRP,**

Periodic Vehicle Routing Problem): Similar al Problema de enrutamiento clásico pero teniendo en cuenta una nueva restricción, hay un período de tiempo estimado para realizar un determinado número de entregas en un mismo punto a lo largo de la ruta. Este problema aparece detallado en el trabajo de B. Golden, S. Raghavan y E. Wasil [26].

Además de las distintas variantes del VRP, el interés del problema ha hecho que con el tiempo hayan surgido también “familias” o grupos de métodos de solución. Se podrían destacar los métodos de solución exactos (basados en formulaciones del problema como un problema de programación lineal entera, en técnicas de búsqueda en árboles y en “Branch-and-cut” y “Branch-and-Bound”) y los heurísticos (basados en encontrar una solución de partida y aplicar un proceso iterativo de mejora de la misma). Como se comenta a continuación, en esta memoria nos vamos a centrar en estudiar algunos de los primeros métodos heurísticos de solución del problema.

Organización de la memoria

En el resto de la memoria vamos a presentar diversos métodos de solución del CVRP. En el Capítulo 1 presentamos el artículo seminal de Dantzig y Ramser de 1959, y su método de solución aproximada. En el Capítulo 2 presentamos el trabajo de Clarke y Wright de 1964, en el que presentan un método heurístico de solución que mejora la solución dada por Dantzig y Ramser. Posteriormente, en el Capítulo 3 haremos una presentación del trabajo de Altinkemer y Gavish [18] de 1991 mostrando un algoritmo de ahorro paralelo basado en el método de Clarke y Wright. Después, en Conclusiones y Trabajo Futuro, mostraremos lo que consideramos lo más importante de toda la memoria, así como un breve resumen de nuestra experiencia realizándola.

Para finalizar, contamos con tres apéndices. En el Apéndice A, mostramos todos los cálculos que hemos realizado siguiendo el algoritmo de Dantzig y Ramser hasta llegar a la solución del ejemplo que mostramos en el Capítulo 1. En el Apéndice B, al igual que en el anterior, mostramos todos los cálculos que hemos realizado siguiendo el algoritmo de Clarke y Wright hasta llegar a la solución del ejemplo que mostramos en el Capítulo 2. En el Apéndice C se explica en qué consiste el problema de emparejamiento de peso máximo (conocido en inglés como “maximum weighted matching problem”), el cual es necesario para resolver el algoritmo presentado en el Capítulo 3.

Capítulo 1

Algoritmo de Dantzig y Ramser

A continuación vamos a presentar el trabajo de Dantzig y Ramser [6] donde se formula el CVRP y se da un algoritmo aproximado de solución del mismo. En el artículo se presenta el algoritmo y un ejemplo de ejecución, pero no se dan demostraciones concretas sobre el nivel de aproximación con respecto a la solución óptima que se consigue.

Lo primero de todo, en la Sección 1.1, vamos a definir una serie de conceptos en los cuales se basa el artículo y que son desarrollados a lo largo de él. Después vamos a mostrar el ejemplo dado en el artículo explicando los pasos que se realizan hasta llegar a una solución aproximada, haciendo una presentación del mismo en la Sección 1.2 y luego presentando el algoritmo en la Sección 1.3. Posteriormente en la Sección 1.4 mostramos un par de problemas similares, los cuales pueden solucionarse con el algoritmo que va a ser presentado. Para finalizar, en el Apéndice A, se encuentran todos los cálculos detallados que hemos realizado para conseguir llegar a la solución del ejemplo que hemos presentado en la Sección 1.2.

1.1. Conceptos básicos

Definición 1. Grafo: Conjunto de objetos unidos por enlaces, que permiten representar relaciones binarias entre elementos de un conjunto (ver Figura 1.1).

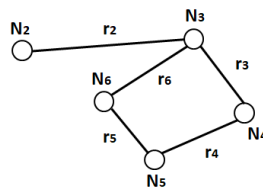


Figura 1.1: Ejemplo de un grafo.

Definición 2. Nodo: También llamado *vértice*, corresponde a los objetos que aparecen en los grafos unidos por enlaces. Por ejemplo, en la Figura 1.1 hay 6 nodos que son N_1, N_2, N_3, N_4, N_5 y N_6 .

Definición 3. Arista: También llamado *arco*, es cada uno de los enlaces que aparecen en un grafo, los cuales corresponden a la relación entre dos nodos. Por ejemplo, en la Figura 1.1 hay 6 aristas que son r_1, r_2, r_3, r_4, r_5 y r_6 .

Definición 4. Nodos adyacentes: Son aquellos nodos que están relacionados, es decir, que están conectados por una arista. Por ejemplo en la Figura 1.1 los nodos N_1 y N_2 son adyacentes porque están relacionados por la arista r_1 .

Definición 5. Punto estacionario: Corresponde a cada uno de los nodos por los cuales es obligatorio que el vehículo pase. Se definen como P_i , siendo $i = 0, 1, \dots, n$ con n el número de puntos estacionarios, que corresponden a los N_i definidos anteriormente.

Definición 6. Punto terminal: También llamado *terminal*, corresponde al primer punto estacionario, del cual empieza la ruta. Se denota P_0 .

Definición 7. Capacidad: Capacidad total que el vehículo lleva y debe entregar a lo largo de toda el grafo. Se denota con la letra C .

Definición 8. Matriz de distancia: Matriz simétrica que especifica la distancia entre todos los puntos estacionarios, es decir, los valores que todas las aristas tienen asociados. Se denota M y cada elemento es $d_{i,j}$ que se refiere a la distancia entre P_i y P_j , considerando por la condición de simetría de la matriz que $d_{i,j} = d_{j,i}$ y por completitud que $d_{i,i} = 0$.

Definición 9. Vector de entrega: Vector que especifica las cantidades que se entregarán en cada punto estacionario. Se denota Q y cada elemento es Q_i que se refiere a la cantidad que se entrega en P_i , respecto a los cuales se asume la siguiente condición:

$$Q_i \leq C, \quad \forall i \in \{1, \dots, n\}. \quad (1.1)$$

Definición 10. Función de coste: También llamada *función objetivo*, es la función que se quiere optimizar (maximizar o minimizar). En nuestro caso representará el coste de recorrer los nodos de un grafo con los distintos vehículos, y por tanto será una combinación lineal de variables enteras a minimizar.

1.2. Presentación del problema

El problema presentado en el artículo de Dantzig y Ramser [6] es el “Problema del reparto de camiones”, el cual los autores consideran como una generalización del “Problema del vendedor-viajero” teniendo en cuenta la capacidad del vehículo anteriormente mencionada. En su forma más sencilla el problema del vendedor ambulante se refiere a la determinación de la ruta más corta que pasa por cada uno de n puntos dados.

Otra generalización de este problema de la que se habla consiste en imponer la condición de que las entregas Q_i especificadas se realicen en cada punto P_i (excepto el terminal). En el caso de que la capacidad total sea

$$C \geq \sum_i Q_i, \quad (1.2)$$

entonces el problema es formalmente idéntico al Problema del vendedor-viajero en su forma original, ya que se puede atender cada punto de entrega en una única ruta que vincule todos los puntos.

Respecto a esto, Dantzig y Ramser comentan que si todos los vehículos tienen la misma capacidad da lo mismo tener un vehículo que tener m vehículos. Esto se debe a que este problema está asociado a minimizar la distancia y hacer todo el recorrido con el mismo vehículo m veces es lo mismo que hacer el mismo recorrido con los m vehículos una vez cada uno, así que vamos a asumir que solo tenemos un vehículo con la capacidad C anteriormente mencionada.

En nuestro caso vamos a tratar el “Problema del reparto de camiones” que se presenta en el artículo considerando un grafo con una serie de nodos a los que denominaremos puntos estacionarios (que tienen un valor asociado que corresponde a la cantidad que se debe descargar en cada uno de ellos), aristas (a las que también se les asocia un valor que corresponde a la distancia con respecto a los otros nodos) y un vehículo (en el artículo se considera un camión) el cual tiene una capacidad y tiene que recorrer el grafo. En este problema la relación entre C y los valores de Q_i se consideran tal que el transportista solo puede realizar entre 1 y t entregas en cada viaje. Así que este problema se caracteriza por la relación

$$C < \sum_i Q_i. \quad (1.3)$$

Por lo que nuestro problema, con lo definido anteriormente, se puede expresar como:

1. Dado un conjunto de n puntos estacionarios P_i con $i = 1, 2, \dots, n$ a los cuales se realizan las entregas desde el punto P_0 , llamado terminal.
2. Se da una matriz de distancia $M = [d_{i,j}]$ que especifica la distancia $d_{i,j} = d_{j,i}$ entre cada par de puntos $\forall i, j = 0, 1, \dots, n$.
3. Se da un vector de entrega $Q = (Q_i)$ que especifica la cantidad Q_i que se entregará en cada punto P_i con $i = 1, 2, \dots, n$.
4. La capacidad del vehículo es C , donde $C > \max Q_i$ ¹.
5. Consideramos que $x_{i,j} = x_{j,i}$, siendo $x_{i,j}$ el valor asociado a la arista que une los nodos N_i y N_j que corresponden con los puntos estacionarios P_i y P_j . Si $x_{i,j} = 1$ significa que la relación entre los puntos P_i y P_j está en

¹Esta condición que dan Dantzig y Ramser no sería necesaria ya que si un punto estacionario tiene una demanda mayor que la capacidad C se podría separar en tantos nodos como se quiera para que en cada uno de ellos haya una demanda menor que C .

la solución para todo $i, j = 0, 1, \dots, n$ y si $x_{i,j} = 0$ significa que la relación entre dichos puntos no está en la solución. De esto se obtiene la condición²:

$$\sum_{j=0}^n x_{i,j} = 1, \quad (1.4)$$

para $i = 1, 2, \dots, n$, ya que cada punto P_i está conectado en la solución con P_0 o, como máximo, con otro punto P_j . Además, por definición, $x_{i,i} = 0$ para cada $i = 0, 1, \dots, n$.

6. El problema es encontrar los valores de $x_{i,j}$ que hacen que la distancia total

$$D = \sum_{i,j=0}^n d_{i,j} x_{i,j} \quad (1.5)$$

sea mínimo bajo las condiciones de la 2 hasta la 5.

El método de solución que proponen Dantzig y Ramser para este problema parte de la idea básica de sintetizar la solución en varias etapas de agregación en las que se realizan suboptimizaciones en pares de puntos o grupos. Estas etapas de agregación consisten en ir haciendo grupos de puntos intentando minimizar la distancia que hay entre ellos partiendo desde una solución de partida, donde todos los puntos están emparejados con el terminal. En la primera agregación se hacen parejas de puntos, en la segunda se juntan las parejas y se hacen grupos de cuatro puntos y así hasta tener las N agregaciones que se consideren, en la cual podría haber grupos de 2^N puntos. El número de etapas de agregaciones que se emplearán se determina de la siguiente manera: se ordenan las entregas, Q_i , en una secuencia $Q_1, Q_2, \dots, Q_i, Q_{i+1}, \dots, Q_n$ tal que $Q_i \leq Q_{i+1}$ para cualquier $i = 1, \dots, n-1$. Entonces se determina t (el cual ha sido nombrado anteriormente y corresponde con el máximo número posible de entregas que se pueden realizar con un vehículo con capacidad C para un conjunto dado de Q_i) tal que

$$\begin{cases} \sum_{i=1}^t Q_i \leq C, \\ \sum_{i=1}^{t+1} Q_i > C. \end{cases} \quad (1.6)$$

Dado que la secuencia Q_1, Q_2, \dots, Q_t representa una combinación factible por la definición dada de t , puede estar en la solución óptima. Esto es así ya que Dantzig y Ramser quieren permitir que en la solución calculada el grupo de nodos P_1, P_2, \dots, P_t , que se pueden recorrer en un único viaje, sea una ruta posible y por tanto el número de agregaciones realizado permita hasta grupos de t puntos. Así que, el método de cálculo del número de agregaciones a emplear debe admitir la combinación Q_1, Q_2, \dots, Q_t en la agregación final. Este número de etapas de agregaciones, N , se determina de tal manera que

$$N \simeq \log_2 t, \quad (1.7)$$

ya que 2^N es el mayor número de puntos que podrían resultar agregados tras N etapas.

²En realidad, lo lógico sería que $\sum_{j=0}^n x_{i,j} = 2$ como veremos en los dos siguientes capítulos, ya que por cada punto entra y sale un vehículo. Sin embargo, Dantzig y Ramser hacen agrupaciones considerando solo las relaciones entre los puntos sin contar con la salida y la llegada desde el terminal. Esas agrupaciones son enlazadas directamente con el terminal al final del problema.

1.3. Algoritmo

1.3.1. Observación general

Según la condición 1.4, $x_{i,j}$ solo puede tomar valores 0 ó 1. En 1959, año en el que fue escrito el artículo, no existía un método de aplicación general para resolver problemas de programación lineal de variables discretas. Sin embargo, sabían determinar la solución si se admite la condición

$$0 \leq x_{i,j} \leq 1, \quad (1.8)$$

y luego aplicando métodos de programación lineal. Esta condición hace posible la aparición de valores fraccionarios en la solución ($0 < x_{i,j} < 1$), lo cual indica la existencia de diferentes alternativas a la hora de emparejar los puntos en las agregaciones, ya que hay que elegir un emparejamiento entre los puntos cuyas relaciones tengan un valor fraccionario. Dantzig y Ramser afirman que la experiencia les ha demostrado que el número de estos emparejamientos alternativos es pequeño, por lo que el emparejamiento que produce el menor kilometraje se puede encontrar fácilmente por prueba y error. Así que si la solución obtenida no satisface el requisito de que $x_{i,j}$ sea 0 ó 1 y fuerzas a que lo cumpla los autores también afirman que el método no garantiza que se obtenga la solución óptima y que es probable que la mejor solución obtenida por este método se aproxime al óptimo a medida que aumenta la cantidad de puntos, debido a que se pueden realizar mayor número de relaciones entre ellos.

1.3.2. Presentación del ejemplo

A continuación, vamos a mostrar el ejemplo que usaron Dantzig y Ramser para explicar el algoritmo. Este ejemplo tiene las siguientes características:

- Hay 12 puntos estacionarios P_1, P_2, \dots, P_{12} además del terminal P_0 .
- La capacidad C es igual a 6000.
- La matriz de distancia simétrica es:

$$M = \begin{bmatrix} 0 & * & * & * & * & * & * & * & * & * & * & * & * \\ 9 & 0 & * & * & * & * & * & * & * & * & * & * & * \\ 14 & 5 & 0 & * & * & * & * & * & * & * & * & * & * \\ 21 & 12 & 7 & 0 & * & * & * & * & * & * & * & * & * \\ 23 & 22 & 17 & 10 & 0 & * & * & * & * & * & * & * & * \\ 22 & 21 & 16 & 21 & 19 & 0 & * & * & * & * & * & * & * \\ 25 & 24 & 23 & 30 & 28 & 9 & 0 & * & * & * & * & * & * \\ 32 & 31 & 26 & 27 & 25 & 10 & 7 & 0 & * & * & * & * & * \\ 36 & 35 & 30 & 37 & 35 & 16 & 11 & 10 & 0 & * & * & * & * \\ 38 & 37 & 36 & 43 & 41 & 22 & 13 & 16 & 6 & 0 & * & * & * \\ 42 & 41 & 36 & 31 & 29 & 20 & 17 & 10 & 6 & 12 & 0 & * & * \\ 50 & 49 & 44 & 37 & 31 & 28 & 25 & 18 & 14 & 12 & 8 & 0 & * \\ 52 & 51 & 46 & 39 & 29 & 30 & 27 & 20 & 16 & 20 & 10 & 10 & 0 \end{bmatrix}$$

- El vector de entrega es:

$$Q = (1200, 1700, 1500, 1400, 1700, 1400, 1200, 1900, 1800, 1600, 1700, 1100).$$

A partir de este vector de entregas y de la capacidad sacamos el número de agregaciones que se van a realizar. Para ello ordenamos el vector:

$$(1100, 1200, 1200, 1400, 1400, 1500, 1600, 1700, 1700, 1700, 1800, 1900).$$

El número máximo de entregas t es igual a 4, ya que:

$$\begin{cases} 1100 + 1200 + 1200 + 1400 = 4900 < 6000, \\ 1100 + 1200 + 1200 + 1400 + 1400 = 6300 > 6000. \end{cases}$$

Por lo que el número de etapas de agregaciones que se van a realizar son:

$$N = \log_2 4 = 2.$$

1.3.3. Primera etapa de agregaciones

Lo primero de todo presentamos al igual que Dantzig y Ramser el método sobre una tabla que incluye las distancias y el vector de entrega (ver Figura 1.2) y definimos cuatro conceptos que se utilizan a lo largo del algoritmo:

Definición 11. Conjunto básico: *Conjunto de las relaciones entre los distintos puntos estacionarios, en el cual se van añadiendo y eliminando las relaciones que finalmente corresponderán a la solución final de tal forma que siempre contiene el mismo número de elementos.*

Definición 12. Entrada básica: *Cada una de las relaciones que están contenidas en el conjunto básico.*

Definición 13. Conjunto no básico: *Conjunto de las relaciones entre los distintos puntos estacionarios, las cuales no han sido añadidas por el momento al conjunto básico o han sido eliminados de él, pudiendo posteriormente formar parte de él.*

Definición 14. Entrada no básica: *Cada una de las relaciones que están contenidas en el conjunto no básico.*

Las entregas se realizan desde el punto P_0 hasta los puntos P_1, \dots, P_{12} . El número en la esquina inferior derecha de cada casilla es la distancia más corta entre los puntos P_i y P_j . Estos números son los elementos de la matriz de distancia descritos en la condición 2. El vector de entrega Q se muestra a la izquierda de la matriz triangular y los $x_{i,j}$ se introducen en la esquina superior izquierda de cada cuadro. Como inicio todos los puntos de entrega P_1, \dots, P_{12} se emparejan con el terminal P_0 para haya 12 entradas $x_{0,i}$ con $i = 1, \dots, 12$. Estas 12 entradas constituyen el conjunto básico al inicio del cálculo (en la imagen las casillas que pertenecen al conjunto básico se muestran en color verde) y durante cada iteración siguiente, exactamente un elemento del conjunto básico se elimina y se reemplaza por un nuevo elemento del conjunto no básico.

Las casillas restantes pertenecen al conjunto no básico y aquellas que están en color rojo muestran las entradas tales que la relación de sus puntos estacionarios no es admisible durante la primera etapa de agregaciones, ya que la demanda combinada de dichos puntos excede la mitad de la capacidad del vehículo. Por ejemplo la demanda combinada de P_4 y P_9 es $1400 + 1800 = 3200$, que es mayor

Q	P0																			
1200	1	9	P1																	
1700	1	14	5	P2																
1500	1	21	12	7	P3															
1400	1	23	22	17	10	P4														
1700	1	22	21	16	21	19	P5													
1400	1	25	24	23	30	28	9	P6												
1200	1	32	31	26	27	25	10	7	P7											
1900	1	36	35	30	37	35	16	11	10	P8										
1800	1	38	37	36	43	41	22	13	16	6	P9									
1600	1	42	41	36	31	29	20	17	10	6	12	P10								
1700	1	50	49	44	37	31	28	25	18	14	12	8	P11							
1100	1	52	51	46	39	29	30	27	20	16	20	10	10	P12						

que $\frac{C}{N} = \frac{C}{2} = 3000$, ya que $N = 2$ es el número de etapas de agregación, y si en la primera etapa se realizan parejas de puntos con capacidades mayores a 3000 entonces esas parejas no podrían emparejarse entre ellas en la siguiente etapa de agregación porque sobrepasarían el valor de C .

Correcciones rápidas

La solución inicial, en la que cada punto está emparejado con el terminal P_0 , puede mejorarse mediante una serie de correcciones rápidas. Dantzig y Ramser afirman que es deseable realizar estas correcciones ya que el número de posteriores iteraciones disminuye a medida que disminuye la diferencia entre la solución inicial y la óptima, aunque no es imprescindible realizar estas correcciones para llegar a la solución. Se pueden hacer correcciones rápidas introduciendo en la solución entradas no básicas que corresponden a valores de $d_{i,j}$ relativamente pequeños, cuyos $d_{0,i}$ y $d_{0,j}$ sean relativamente grandes. Por ejemplo, $d_{6,7} = 7$ es relativamente pequeño y puede introducirse en el conjunto básico ingresando con el valor de una variable α en la esquina superior izquierda correspondiente. Para satisfacer la condición 1.4, las sumas de los $x_{i,j}$ para cualquier $i = 1, 2, \dots, n$ debe ser igual a 1. Dado que la matriz de distancia M es simétrica, se deduce que la suma de los valores $x_{i,j}$ de la fila 6 y la columna 6 en la matriz triangular que se muestra en la Figura 1.2 también debe ser igual a 1, y lo mismo para la fila 7 y la columna 7. Por lo tanto, se hacen las siguientes correcciones:

$$\begin{cases} x_{6,7} = \alpha, \\ x_{0,6} = 1 - \alpha, \\ x_{0,7} = 1 - \alpha, \end{cases} \quad (1.9)$$

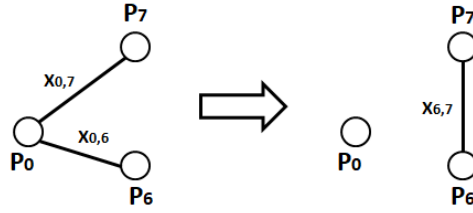


Figura 1.3: Corrección rápida geométrica.

siendo 1 el máximo valor que α puede alcanzar debido a la condición 1.8, que haría que:

$$\begin{cases} x_{6,7} = 1, \\ x_{0,6} = 0, \\ x_{0,7} = 0, \end{cases} \quad (1.10)$$

por lo que Dantzig y Ramser deciden tomar $\alpha = 1$.

Dado que se debe eliminar una entrada básica, hay que elegir entre $x_{0,6}$ y $x_{0,7}$. Como la distancia $d_{0,7}$ es mayor que $d_{0,6}$, la entrada básica $x_{0,7}$ se elimina del conjunto básico para reducir la distancia total tanto como sea posible. Los cambios realizados anteriormente son que la entrada $x_{6,7} = 1$ se ha agregado al conjunto básico, que el valor de $x_{0,6}$ ha cambiado de 1 a 0 y que se ha eliminado $x_{0,7}$, manteniendo así el mismo número de entradas básicas en el conjunto básico. Geométricamente, esto significa que los puntos 6 y 7 se han separado del terminal y se han unido entre sí (ver Figura 1.3).

Este proceso de hacer correcciones rápidas se repite siempre que alguna entrada no básica con un valor relativamente pequeño³ de $d_{i,j}$ esté disponible. Además, afirman que una buena regla para obtener una mejora rápida de la solución es omitir los valores de $d_{i,j}$ si las entradas $x_{0,i}$ o $x_{0,j}$ ya se han eliminado del conjunto básico o tienen el valor 0. A partir de la Figura 1.2, las primeras cuatro iteraciones son las siguientes:

- **Iteración 1:** como $d_{1,2} = 5$ y $d_{0,1} = 9 < d_{0,2} = 14$:
 - $x_{1,2} = 1$ reemplaza $x_{0,2}$.
 - $x_{0,1} = 1$ pasa a ser $x_{0,1} = 0$.
- **Iteración 2:** como $d_{6,7} = 7$ y $d_{0,6} = 25 < d_{0,7} = 32$:
 - $x_{6,7} = 1$ reemplaza $x_{0,7}$.
 - $x_{0,6} = 1$ pasa a ser $x_{0,6} = 0$.
- **Iteración 3:** como $d_{3,4} = 10$ y $d_{0,3} = 21 < d_{0,4} = 23$:
 - $x_{3,4} = 1$ reemplaza $x_{0,4}$.

³Se suelen escoger aquellos números tales que $d_{i,j} \ll d_{0,i}$ y $d_{i,j} \ll d_{0,j}$.

[illegible]

- $x_{0,3} = 1$ pasa a ser $x_{0,3} = 0$.

- **Iteración 4:** como $d_{11,12} = 10$ y $d_{0,11} = 50 < d_{0,12} = 52$:
- $x_{11,12} = 1$ reemplaza $x_{0,12}$.
 - $x_{0,11} = 1$ pasa a ser $x_{0,11} = 0$.

Tras las correcciones rápidas, queda la tabla que se ve en la Figura 1.4 donde se puede observar el nuevo conjunto básico (casillas de color verde). Además, como resultado de las primeras cuatro correcciones rápidas, la suma de las distancias entre pares se ha reducido de 364 a 170 unidades, contando en ambos casos los $d_{i,j}$ asociados a las entradas básicas con $x_{i,j} = 1$.

La función Delta

A medida que se van introduciendo a la solución un mayor número de parejas de puntos con pequeñas distancias entre ellos, la distancia total se irá disminuyendo y la diferencia con la solución óptima será también más pequeña, así que para ir desechando posibilidades hay que calcular la distancia total en cada caso. Esto realizado en un procedimiento de prueba y error requeriría una enorme cantidad de cómputo incluso si el número de puntos no fuera demasiado grande.

Por lo tanto, es necesario un criterio que permita aceptar o rechazar una entrada no básica. Este criterio es proporcionado por una función de costes que da información sobre cómo se modifica la distancia total D al introducir una entrada no básica en el conjunto básico, denominada función Delta, que está definida como

$$\delta_{i,j}^{(n)} = \pi_i^{(n)} + \pi_j^{(n)} - d_{i,j}, \quad (1.11)$$

donde $\pi_i^{(n)}$ y $\pi_j^{(n)}$ son constantes determinadas en la n -ésima iteración. Por definición, $\pi_i^{(n)}$ y $\pi_j^{(n)}$ están determinadas para que

$$\delta_{i,j}^{(n)} = 0 \quad (1.12)$$

para todos los $d_{i,j}$ correspondientes a las entradas básicas, mientras que $\delta_{i,j}^{(n)} \neq 0$ para las entradas no básicas⁴.

En el caso de que $\delta_{i,j}^{(n)} > 0$, la distancia total D disminuye si se introduce la entrada no básica $x_{i,j}$ en el conjunto básico, mientras que si $\delta_{i,j}^{(n)} < 0$, la distancia total D aumenta para el $x_{i,j}$ correspondiente. Por lo tanto, hay que ir introduciendo entradas no básicas mientras haya algún $\delta_{i,j}^{(n)} > 0$, en caso contrario ya se habrá llegado a la mejor solución.

Sin embargo, siempre que al menos un $\delta_{i,j}^{(n)} > 0$, es posible una reducción de la distancia total. Si varios $\delta_{i,j}^{(n)}$ son mayores que 0, Dantzig y Ramser optan por seleccionar la mayor reducción posible que es la correspondiente al $\delta_{i,j}^{(n)}$ de mayor valor y si dos $\delta_{i,j}^{(n)}$ son iguales se quedan con el primero y descartan el segundo para evitar arbitrariedades. En el caso que $\delta_{i,j}^{(n)} = 0$ para una entrada no básica, Dantzig y Ramser deciden quedarse con la entrada que menor $d_{i,j}$ tenga.

Las constantes $\pi_i^{(n)}$ y $\pi_j^{(n)}$ de la función Delta 1.11 pueden determinarse a partir de la condición 1.12, dando lugar a la ecuación:

$$\pi_i^{(n)} + \pi_j^{(n)} - d_{ij} = 0. \quad (1.13)$$

Los autores establecen arbitrariamente $\pi_0 = 0$; además, afirman que cualquier elección de un número entero mayor o igual que 0 es admisible y produciría el mismo resultado (aunque no demuestran tal afirmación). Como en nuestro ejemplo hay 12 entradas básicas tenemos 12 ecuaciones similares a la ecuación 1.13 a partir de las cuales se pueden determinar los 12 valores π_i correspondientes a cada punto de entrega P_i de la siguiente forma:

$$x_{0,1}) : \pi_0 + \pi_1 - d_{0,1} = 0 \implies \pi_1 - 9 = 0 \implies \boxed{\pi_1 = 9}$$

$$x_{1,2}) : \pi_1 + \pi_2 - d_{1,2} = 0 \implies 9 + \pi_2 - 5 = 0 \implies \boxed{\pi_2 = -4}$$

$$x_{0,3}) : \pi_0 + \pi_3 - d_{0,3} = 0 \implies \pi_3 - 21 = 0 \implies \boxed{\pi_3 = 21}$$

$$x_{3,4}) : \pi_3 + \pi_4 - d_{3,4} = 0 \implies 21 + \pi_4 - 10 = 0 \implies \boxed{\pi_4 = -11}$$

$$x_{0,5}) : \pi_0 + \pi_5 - d_{0,5} = 0 \implies \pi_5 - 22 = 0 \implies \boxed{\pi_5 = 22}$$

$$x_{0,6}) : \pi_0 + \pi_6 - d_{0,6} = 0 \implies \pi_6 - 25 = 0 \implies \boxed{\pi_6 = 25}$$

$$x_{6,7}) : \pi_6 + \pi_7 - d_{6,7} = 0 \implies 25 + \pi_7 - 7 = 0 \implies \boxed{\pi_7 = -18}$$

$$x_{0,8}) : \pi_0 + \pi_8 - d_{0,8} = 0 \implies \pi_8 - 36 = 0 \implies \boxed{\pi_8 = 36}$$

⁴La explicación intuitiva de esto es que para las entradas básicas se tiene que $\delta_{i,j}^{(n)} = 0$ porque incluirlas en la solución no modificaría el coste total de la solución (ya forman parte de ella), mientras que para las entradas no básicas $\delta_{i,j}^{(n)} \neq 0$ porque introducirlas aumentará o reducirá el coste de la solución.

Figura 1.5: Tabla de los correspondientes valores de la funciones Delta al inicio de la primera etapa de agregaciones

$$x_{0,9}) : \pi_0 + \pi_9 - d_{0,9} = 0 \implies \pi_9 - 38 = 0 \implies \boxed{\pi_9 = 38}$$

$$x_{0,10}) : \pi_0 + \pi_{10} - d_{0,10} = 0 \implies \pi_{10} - 42 = 0 \implies \boxed{\pi_{10} = 42}$$

$$x_{0,11}) : \pi_0 + \pi_{11} - d_{0,11} = 0 \implies \pi_{11} - 50 = 0 \implies \boxed{\pi_{11} = 50}$$

$$x_{11,12}) : \pi_{11} + \pi_{12} - d_{11,12} = 0 \implies 50 + \pi_{12} - 10 = 0 \implies \boxed{\pi_{12} = -40}$$

Estos valores junto con π_0 se ingresan en la esquina inferior derecha de los cuadros que contienen las identificaciones de los puntos estacionarios como se muestra en la Figura 1.5.

Una vez calculados todos los π_i se calculan todos los $\delta_{i,j}^{(1)}$ (ver Figura 1.5) de todas las entradas a partir de la ecuación (1.11 y se selecciona el de mayor valor (casilla de color amarillo en la Figura 1.5). Entonces, se establece dicho $x_{i,j}$ como θ y el resto de las entradas básicas se corrigen de tal manera que $\sum_{i=0}^n x_{i,j} = 1$ con $j = 6, 10$ (ver Figura 1.6).

En el ejemplo si hacemos todos los $\delta_{i,j}^{(1)}$, vemos que hay más de uno mayor que cero:

$$\delta_{1,3} = \pi_1^{(1)} + \pi_3^{(1)} - d_{1,3} = 9 + 21 - 12 = 18,$$

$$\delta_{1,5} = \pi_1^{(1)} + \pi_5^{(1)} - d_{1,5} = 9 + 22 - 21 = 10,$$

$$\delta_{1,6} = \pi_1^{(1)} + \pi_6^{(1)} - d_{1,6} = 9 + 25 - 24 = 10,$$

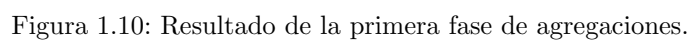
$$\delta_{1,9} = \pi_1^{(1)} + \pi_9^{(1)} - d_{1,9} = 9 + 38 - 37 = 10,$$

$$\delta_{1,10} = \pi_1^{(1)} + \pi_{10}^{(1)} - d_{1,10} = 9 + 42 - 41 = 10,$$

$$\delta_{1,11} = \pi_1^{(1)} + \pi_{11}^{(1)} - d_{1,11} = 9 + 50 - 49 = 10,$$

$$\delta_{3,6} = \pi_3^{(1)} + \pi_6^{(1)} - d_{3,6} = 21 + 25 - 30 = 6,$$

Figura 1.9: Tabla de los correspondientes valores de la funciones Delta al final de la primera etapa de agregaciones.



A0									
1	28	A1							
1	54	54	A2						
1	44	52	72	A3					
1	84	92	102	84	A4				
1	86	94	110	86	92	A5			
1	72	80	102	74	84	86	A6		
1	112	120	120	112	112	112	112	A7	

A0
28 A1
54 W2 54 A2
44 W2 52 W2 72 A3
84 92 102 84 A4
86 94 110 86 0 92 A5
72 80 102 74 1 84 86 A6
112 120 120 112 112 1 112 112 A7

Tratamiento fraccionario de $x_{i,j}$

En este diagrama A_1 , A_2 y A_3 se colocan en los vértices de un triángulo y A_0 se coloca en el centro, donde se conecta con el resto de los vértices. Las distancias correspondientes a las 6 posibles agregaciones se toman de la Figura 1.12 y se colocan en las líneas correspondientes de la Figura 1.13. Se ve fácilmente que las agregaciones que producen la distancia más pequeña son $(A_1A_2) + (A_0A_3)$, ya que $d_{A_0,A_3} + d_{A_1,A_2} = 44 + 54 = 98$, mientras que otras posibilidades que existen son $(A_0A_1) + (A_2A_3)$ y $(A_0A_2) + (A_1A_3)$ cuyas distancias suman 100 y 106 respectivamente. Como ampliación, en el caso de que el número de entradas fraccionarias fuera 5, los símbolos A_i se colocan en los vértices de un polígono de cinco lados y A_0 se colocaría dentro, y así con todos los números impares posibles (si fueran 7 un heptágono, si fueran 9 un eneágono...).

Sin embargo, si hay uno o más grupos impares de valores fraccionarios al final de la segunda etapa o superior, Dantzig y Ramser suponen que es mejor seleccionar arbitrariamente uno de los grupos, determinar la agregación óptima para el grupo y luego volver a calcular la etapa después de que los A'_i s pares con $i \neq 0$ se hayan eliminado de la matriz⁵. Por lo tanto, en el ejemplo que estamos solucionando, se recomienda que la segunda etapa se resuelva de nuevo

⁵La explicación intuitiva de esto es que al haber hecho agrupaciones impares, el punto que se descarta al formar las parejas por medio de la Figura 1.13 puede mejorar la solución si se empareja con otro punto ajeno a la agrupación impar.

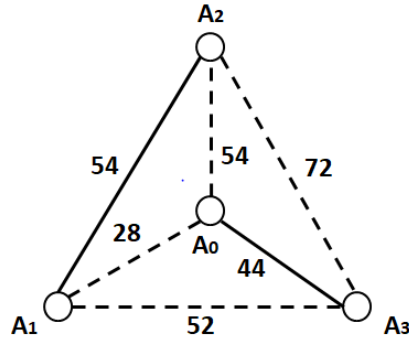


Figura 1.13: Diagrama con las relaciones.

A0							
1	44	A3					
	84		84	A4			
	86		86	0	92	A5	
	72	0	74	1	84	86	A6
	112	112	112	1	112	112	A7

Figura 1.14: Tabla durante el tratamiento fraccionarios de los $x_{i,j}$.

asumiendo que A_1 y A_2 están agregados, pero que A_3 , A_4 , A_5 , A_6 y A_7 son libres de emparejarse como deseen (partimos desde la tabla de la Figura 1.14).

Resolviendo de la misma forma que hemos resuelto el resto de tablas tras una sola iteración llegamos a la tabla que se muestra en la Figura 1.15, en la que nuevamente hay un grupo impar que involucra los pares de puntos A_4 , A_5 y A_7 . Por el procedimiento ilustrado en la Figura 1.13, vemos que A_5 debe agregarse con A_7 (ya que la distancia que se consigue con $(A_0A_4) + (A_5A_7)$ es 196 mientras que con otras relaciones se consiguen 198 y 204). Finalmente, A_3 , A_4 y A_6 se resuelven con el mismo método, de donde se saca que la mejor agregación es $(A_0A_3) + (A_4A_6)$ con una distancia de 128 (ver Apéndice A).

1.3.5. Comparación de la solución final con la óptima

La solución final del ejemplo que resuelven Dantzig y Ramser es (A_1A_2) , (A_5A_7) , (A_4A_6) , A_3 . De acuerdo con la Figura 1.10, la solución con relación a

A0							
44	A3						
84		84	A4				
86		86	1/2	92	A5		
72	1	74	0	84	86	A6	
112	112	112	1/2	112	112	112	A7

Figura 1.15: Tabla de ejemplo tras el primer tratamiento fraccionario de $x_{i,j}$.

los puntos estacionarios es $(P_1P_2P_3P_4)$, $(P_7P_{12}P_{11}P_9)$, $(P_6P_{10}P_8)$, P_5 . Según la Figura 1.12, la distancia total cubierta en este viaje es de 294 unidades, y los autores creían que la asignación $(P_1P_2P_3P_4)$, $(P_7P_{12}P_{11}P_{10})$, $(P_6P_8P_9)$, P_5 con una distancia total de 290 unidades era la solución óptima de este problema. Debido a esa suposición suya, terminan el artículo afirmando que el algoritmo desarrollado en este documento ha dado como resultado una solución que se acerca mucho al óptimo para el ejemplo que ha sido desarrollado. Además Dantzig y Ramser afirman que la experiencia con el método les ha demostrado que se pueden obtener resultados similares en otros casos numéricos y que la diferencia entre la distancia de la solución elegida y la óptima disminuye a medida que aumenta el número de puntos estacionarios.

1.4. Problemas relacionados

Como hemos comentado anteriormente el número de vehículos no es un dato relevante en el problema cuando todos ellos tienen la misma capacidad. Así que todos los problemas de n vehículos cuya capacidad es la misma se resuelven igual que el presentado en este capítulo, sea n el número que sea, ya que se considera que un solo vehículo puede realizar él solo los n viajes sin suponer ningún coste extra.

Sin embargo, existe otro problema que trata sobre vehículos de diferentes capacidades, del cual Dantzig y Ramser afirman que se puede usar el mismo modelo matemático que acaba de ser mostrado, debido a que se puede asumir que se realizan todos los viajes con el vehículo de mayor capacidad, ya que en este algoritmo solo se tiene en cuenta los costes relacionados a la distancia recorrida.

Capítulo 2

Algoritmo de Clarke y Wright

A continuación vamos a presentar el trabajo de Clarke y Wright [5] donde se propone un método de solución aproximado que mejore la anteriormente explicada en el Capítulo 1 por medio de una heurística y de una función de costes. En este artículo los autores intentan encontrar de forma aproximada un enrutamiento para un conjunto de vehículos intentando acercarse lo más posible al enrutamiento óptimo. Estos vehículos, que al contrario del capítulo anterior tienen distintas capacidades, son utilizados para la entrega desde un depósito central, el terminal, a una gran cantidad de puntos de entrega, denominados puntos estacionarios. Se desea asignar cargas a estos vehículos de tal manera que se entregue toda la mercancía total recorriendo la menor distancia total posible.

El capítulo¹ lo separamos en cuatro secciones. En la Sección 2.1, vamos a definir una serie de conceptos en los cuales se basa el artículo, que se añaden a los definidos en el Capítulo 1 y que son desarrollados a lo largo de él. En la Sección 2.2, vamos a hacer una presentación del problema comparándolo con lo visto en el anterior capítulo y mostrando una serie de aspectos teóricos que necesitan una previa formulación del problema para poder entender lo que se va a explicar posteriormente. En la Sección 2.3 explicamos el procedimiento computacional del algoritmo que usaban Clarke y Wright para resolver el problema. En la Sección 2.4 vamos a mostrar algunas mejoras de este algoritmo que se han ido realizando en artículos posteriores. Para finalizar, en el Apéndice B, se encuentran todos los cálculos detallados que hemos realizado para conseguir llegar a la solución del ejemplo que hemos presentado en el Capítulo 1 por medio del algoritmo de Clarke y Wright.

¹A lo largo de este capítulo hemos cambiado la notación y nombres que propusieron los autores para mantener la misma notación que hemos usado en el capítulo anterior.

2.1. Conceptos básicos

Definición 15. Ahorro: Mejora que se obtiene respecto a una solución inicial sobre la función de coste.

Definición 16. Celda ($y:z$): Celda correspondiente a la arista $r_{y,z}$ que está entre dos puntos P_y y P_z con $y, z \in \{0, 1, \dots, m\}$ siendo m el número total de puntos estacionarios, que contiene toda la información relacionada a dicha arista.

2.2. Presentación del problema

2.2.1. Formulación del problema

Tenemos un conjunto de vehículos v_i de capacidad C_i con $i = 1, \dots, n$ y se requiere que las cargas Q_j se entreguen a una serie de puntos P_j con $j = 1, \dots, m$ desde un depósito o terminal P_0 . Dadas las distancias $d_{y,z}$ entre todos estos puntos, se requiere minimizar la distancia total recorrida por los vehículos. Para comodidad del cálculo, ordenamos los C_i tal que

$$C_{i-1} < C_i, \quad \forall i \in \{2, \dots, n\}, \quad (2.1)$$

y asumimos que

$$C_1 < \sum_{j=1}^m Q_j. \quad (2.2)$$

En el caso de que en la solución algunos camiones solo puedan cargarse parcialmente, v_i debe ser lo suficientemente grande como para garantizar que todas las cargas estén asignadas y esa es la motivación que llevó a los autores a realizar este algoritmo².

2.2.2. Comparación con el método de Dantzig y Ramser

En el método explicado anteriormente de Dantzig y Ramser existía la restricción de que en las primeras N etapas solo los puntos cuya carga combinada no exceda de $\frac{C_1}{2^{N-r}}$ pueden unirse en la r etapa de agregación. En el trabajo de Clarke y Wright se elimina esta restricción permitiendo que en cada etapa puedan unirse cualesquiera dos puntos cuya carga combinada no exceda de C_1 . Esto lo hacemos ya que Clarke y Wright afirman que este método tras varios casos probados da mejores resultados que el método de Dantzig y Ramser, debido a una serie de aspectos entre los que destaca el siguiente:

En las primeras etapas de agregación al limitar los emparejamientos a aquellos que no superan una determinada carga puede que los emparejamientos que se realicen tengan grandes distancias entre sus respectivos puntos.

²Si $C_N \geq \sum_{j=1}^m Q_j$ entonces el problema se podría resolver por medio de un TSP.

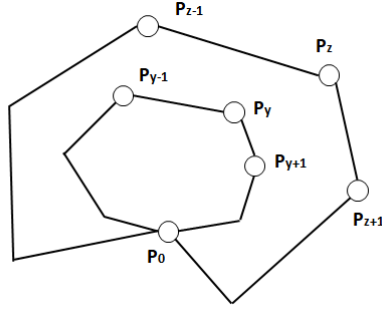


Figura 2.1: Posiciones factibles.

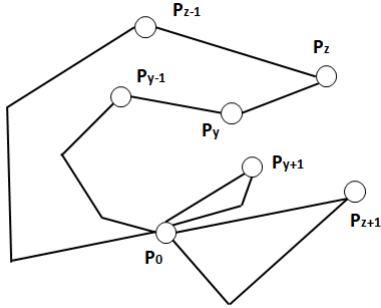


Figura 2.2: 1ª descomposición.

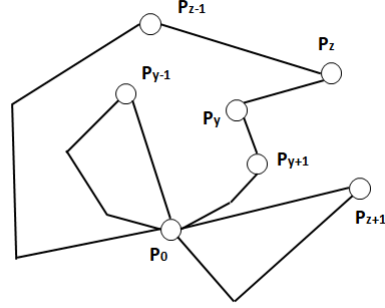


Figura 2.3: 2ª descomposición.

2.2.3. Aspectos teóricos del problema

Consideramos que cada punto está relacionado con otros dos puntos, de los cuales uno o ambos pueden ser el terminal P_0 . Consideramos los dos puntos P_y y P_z con sus respectivos puntos relacionados que son $P_{y\pm 1}$ y $P_{z\pm 1}$, pero asumiendo que P_y y P_z no están vinculados en la misma relación con P_0 . La Figura 2.1 muestra las posiciones de P_y y de P_z en una asignación factible.

Para poder calcular el ahorro derivado de visitar con el mismo vehículo los puntos P_y y P_z , Clarke y Wright usan la representación gráfica que mostramos en las Figuras 2.2, 2.3, 2.4 y 2.5. Estas posibles descomposiciones generan una serie de ahorros:

- $d_{y,y+1} - d_{0,y+1} + d_{z,z+1} + d_{0,z+1} + d_{y,z}$ (ver Figura 2.2),
- $d_{y-1,y} - d_{0,y-1} + d_{z,z+1} + d_{0,z+1} + d_{y,z}$ (ver Figura 2.3),
- $d_{y,y+1} - d_{0,y+1} + d_{z,z-1} + d_{0,z-1} + d_{y,z}$ (ver Figura 2.4),
- $d_{y-1,y} - d_{0,y-1} + d_{z,z-1} + d_{0,z-1} + d_{y,z}$ (ver Figura 2.5).

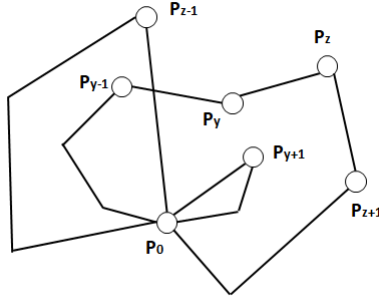


Figura 2.4: 3ª descomposición.

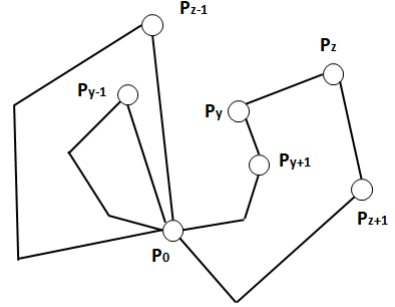


Figura 2.5: 4ª descomposición.

Una vez calculados los cuatro posibles ahorros de cada relación se selecciona el máximo de estos cuatro. Además, cada vez que un punto se agrupa con otros dos (siempre que no sea con P_0), todos sus ahorros se vuelven negativos, y este punto no se volverá a considerar para el enlace. Esto último no queda muy claro en el artículo original por lo que hemos decidido explicarlo más detalladamente. Para ello tomamos cuatro puntos cualesquiera P_1, P_2, P_3 y P_4 , donde suponemos que todos los puntos están unidos al terminal en la primera solución factible y su coste es:

$$D_0 = 2 \cdot (d_{0,1} + d_{0,2} + d_{0,3} + d_{0,4}).$$

A continuación calculamos los ahorros que obtenemos al hacer una pareja de dos de estos puntos separándolos del terminal P_0 :

$$P_1 P_2 \Rightarrow S_{1,2} = d_{0,1} + d_{0,2} - d_{1,2},$$

$$P_1 P_3 \Rightarrow S_{1,3} = d_{0,1} + d_{0,3} - d_{1,3},$$

$$P_1 P_4 \Rightarrow S_{1,4} = d_{0,1} + d_{0,4} - d_{1,4},$$

$$P_2 P_3 \Rightarrow S_{2,3} = d_{0,2} + d_{0,3} - d_{2,3},$$

$$P_2 P_4 \Rightarrow S_{2,4} = d_{0,2} + d_{0,4} - d_{2,4},$$

$$P_3 P_4 \Rightarrow S_{3,4} = d_{0,3} + d_{0,4} - d_{3,4}.$$

Ahora suponemos que el máximo ahorro, $\max\{S_{i,j} \mid i, j \in \{1, 2, 3, 4\}\}$, se ha conseguido con los puntos P_i y P_j con $i, j \in \{1, 2, 3, 4\}$, teniendo un coste de:

$$D_1 = D_0 - \max\{S_{i,j} \mid i, j \in \{1, 2, 3, 4\}\}.$$

El siguiente paso es seleccionar otro punto P_k con $k \in \{1, 2, 3, 4\}$ y $k \neq i, j$, y recalculamos los ahorros si lo metiéramos en el conjunto sacando a P_i o a P_j :

$$P_k(P_i P_j) \text{ sacando } P_i \Rightarrow S_{i,k} - \max\{S_{i,j} \mid i, j \in \{1, 2, 3, 4\}\} \leq 0,$$

$$P_k(P_i P_j) \text{ sacando } P_j \Rightarrow S_{j,k} - \max\{S_{i,j} \mid i, j \in \{1, 2, 3, 4\}\} \leq 0.$$

Vemos que en todos los casos los ahorros son negativos y como resultado de esto Clarke y Wright deciden seleccionar $d_{0,y} + d_{0,z} - d_{y,z}$ como el máximo ahorro asociado a la celda $(y : z)$.

Q	P0																	P12						
1200	2 9	P1																						
1700	2 14	18	5	P2																				
1500	2 21	18	12	28	7	P3																		
1400	2 23	10	22	20	17	34	10	P4																
1700	2 22	10	21	20	16	22	21	26	19	P5														
1400	2 25	10	24	16	23	16	30	20	28	38	9	P6												
1200	2 32	10	31	20	26	26	27	30	25	44	10	50	7	P7										
1900	2 36	10	35	20	30	20	37	24	35	42	16	50	11	58	10	P8								
1800	2 38	10	37	16	36	16	43	20	41	38	22	50	13	54	16	68	6	P9						
1600	2 42	10	41	20	36	32	31	36	29	44	20	50	17	64	10	72	6	68	12	P10				
1700	2 50	10	19	20	44	34	37	42	31	44	28	50	25	64	18	72	14	76	12	84	8	P11		
1100	2 52	10	51	20	46	34	39	46	29	44	30	50	27	64	20	72	16	70	20	84	10	92	10	P12

VEHÍCULO	≤ 4000	> 4000	> 5000	> 6000	DISTANCIA MINIMA	728
DISPONIBLES	∞	7	4	0		
ASIGNADOS	12	0	0	0		

Figura 2.6: Tablas del ejemplo correspondientes al principio del algoritmo.

2.3. Procedimiento computacional

En primer lugar los autores asumen que los valores de Q_j son tales que es posible una asignación inicial de un vehículo a cada punto estacionario. Si esto no es cierto, se puede hacer una asignación dividiendo una carga en dos (o más) vehículos completos de las capacidades más altas disponibles y considerando el resto de esa carga como una cantidad menor. Por ejemplo, si $Q_j = 1700$ y los vehículos disponibles tienen las siguientes capacidades: uno de 700, dos de 600, tres de 500, entonces los vehículos de 700 y 600 se asignarían a ese punto estacionario y faltarían 400 que vendrían de uno de los tres vehículos de 500. Por tanto podemos concluir:

$$Q_i \leq C_n, \quad \forall i \in \{1, \dots, m\}. \quad (2.3)$$

Para el cálculo vamos a usar una matriz tal y como hacen Clarke y Wright, que también usan en su trabajo el mismo ejemplo numérico que habían usado Dantzig y Ramser. En la primera tabla de la Figura 2.6, las entradas en la esquina inferior derecha de cada celda ($y : z$) son las distancias más cortas posibles entre dichos puntos, representado como $S_{y,z}$, mientras que las entradas en la esquina inferior izquierda de cada celda son los ahorros, calculados de forma que para la celda ($y : z$), con $y, z \geq 1$ ($y \neq z$) su valor sea $d_{0,y} + d_{0,z} - d_{y,z}$. Además se incluye el vector de columna $Q = (Q_1, \dots, Q_m)$ en el lado izquierdo de la matriz como hacían Dantzig y Ramser..

Se podría decir que este algoritmo está basado en las cargas Q_j que son requeridas por cada punto P_j con $j = 1, \dots, m$, ya que para poder seguir con el algoritmo siempre debe haber dos grupos de puntos cuya demanda conjunta no sobrepase la capacidad total. La entrada de la esquina superior izquierda de la celda $(y : z)$ corresponde al valor $x_{y,z}$, el cual puede tomar diferentes valores. Si

los dos clientes P_y y P_z están vinculados en la ruta de un vehículo $x_{y,z} = 1$, en caso contrario $x_{y,z} = 0$ para $y, z > 0$. En el caso de que un punto estacionario sea atendido exclusivamente por un vehículo entonces $x_{y,0} = 2$. Además, siempre para todo $y \in \{1, 2, \dots, m\}$ existe la siguiente condición³:

$$\sum_{z=0}^{y-1} x_{z,y} + \sum_{z=y+1}^m x_{y,z} = 2. \quad (2.4)$$

Esta condición muestra que que en este artículo, al contrario que en el anterior, sí se tiene en cuenta que cada vehículo entra y sale por cada punto estacionario. Esto provoca que una vez calculadas las agrupaciones ya se tiene la solución final, al contrario a Dantzig y Ramser que tenían que juntar esas agrupaciones con el terminal para conseguir la solución final.

Para empezar, la solución básica inicial se considera $x_{y,0} = 2$ para todo $y = 1, \dots, m$, es decir, se consideran que todos los puntos estacionarios están únicamente relacionados con el terminal P_0 , habiendo una distancia mínima de 728, la cual como acabamos de señalar corresponde a la distancia de la solución final. En la segunda tabla de la Figura 2.6 se muestra el número de vehículos disponibles clasificados por su capacidad dependiendo en qué rango⁴ se encuentren y el número de vehículos ya asignados.

En el ejemplo numérico que tratamos, se supone que existe un número ilimitado de vehículos con capacidad de 4000, tres vehículos con capacidad de 5000 y cuatro vehículos con capacidad de 6000. Así que en la Figura 2.6 se muestra la solución factible inicial.

Una vez hecho esto se busca en las filas y columnas de la tabla el ahorro máximo (mostrado en color verde en la primera tabla de la Figura 2.6), el cual haga que se respeten las siguientes condiciones:

1. $x_{y,0}$ y $x_{z,0}$ deben ser mayores que cero.
2. P_y y P_z no pueden estar asignados en la misma ruta de un vehículo.
3. Tras eliminar los vehículos asignados a los conductores Q_y y Q_z y agregar un vehículo para cubrir la carga $Q_y + Q_z$ los vehículos asignados no exceden los vehículos disponibles en ninguna columna de la segunda tabla de la Figura 2.6.

Para facilitar el cálculo, las celdas que no cumplan estas condiciones se muestran en color rojo, al igual que todas las celdas de la forma $(y : 0)$, ya que ellas no pueden añadirse como nuevas relaciones. En el caso de que se cumplan estas condiciones, $x_{y,z}$ se hace igual a 1 y otros valores de $x_{i,j}$ son modificados de tal forma que se cumpla la condición 2.4. El vector Q se modifica haciendo primero que todos los Q_j sean cero donde $x_{j,0}$ es cero y haciendo que Q_j sea igual a la carga total en el recorrido para todos los demás j , es decir se le asigna el valor total de la carga acumulada en el trayecto al pasar por ambos puntos,

³En el artículo aparece la condición $\sum_{z=0}^{y-1} x_{y,z} + \sum_{z=y+1}^m x_{y,z} = 2$, la cual está mal.

⁴Los cuatro rangos que se contemplan son: menores o iguales que 4000, mayores que 4000, mayores que 5000 y mayores que 6000, los cuales se superponen entre sí.

Figura 2.7: Tablas en la segunda iteración.

Segunda iteración

$$Q_{11}^2 = Q_{12}^2 = Q_{11}^1 + Q_{12}^1 = 1700 + 1100 = 2800.$$

Tercera iteración

Al igual que en la anterior iteración, se cambia a uno el valor de la celda (10 : 12), y se reajustan los valores $x_{0,10}$ y $x_{0,12}$ disminuyendo en una unidad. Esto hace que $x_{0,12} = 0$, por lo que por la condición 1 todos las celdas de la forma ($y : 12$) ya no van a poder ser seleccionadas por lo que las mostramos de color rojo en la primera tabla de la Figura 2.8. Como la celda (10 : 11),

Q	P0																
5800	1 9																
-	14	1 18	5														
-	21	18	12	1 28	7												
5800	1 23	10	22	20	17	1 34	10										
1700	2 22	10	21	20	16	22	21	26	19								
5100	1 25	10	24	16	23	16	30	20	28	38	9						
5600	1 32	10	31	20	26	26	27	30	25	44	10	50	7				
-	36	10	35	20	30	20	37	24	35	42	16	50	11	58	10		
5100	1 38	10	37	16	36	16	43	20	41	38	22	50	13	54	16	1 68	6
5600	1 42	10	41	20	36	32	31	36	29	44	20	50	17	64	10	72	6
-	50	10	19	20	44	34	37	42	31	44	28	50	25	64	18	72	14
-	52	10	51	20	46	34	39	46	29	44	30	50	27	64	20	72	16

Figura 2.9: Tablas al final del algoritmo.

dando una distancia total de 290 unidades, que coincide con el valor que Dantzig y Ramser consideraban como el óptimo, como hemos comentado anteriormente en el Capítulo 1.

Para finalizar cabe destacar que aunque la mejora en este ejemplo es leve, en un ejemplo con 30 puntos estacionarios Clarke y Wright afirman que obtuvieron una mejora del 17 % respecto al método de Dantzig y Ramser (los resultados correspondientes se encuentran junto a la matriz de kilometraje total en los apéndices del trabajo de Clarke y Wright [5]). Además, los autores afirman que su solución define el orden en que se deben visitar los puntos, pero podría ser beneficioso realizar un problema del vendedor-viajante (TSP) para cada vehículo una vez encontrado el conjunto de puntos a recorrer para intentar llegar a una solución mejor. Finalmente, dicen que en su algoritmo existe una serie de limitaciones prácticas, como la existencia de ciertos puntos que solo aceptan algunos tipos de vehículos, así como otros tratamientos prioritarios que según Clarke y Wright pueden incorporarse al cálculo sin mucha dificultad.

2.4. Mejoras del método de solución

A continuación vamos a comentar algunos de los algoritmos heurísticos que Altinel y Oncan [12] han reunido y presentado en un artículo, los cuales han intentado mejorar el algoritmo de Clarke y Wright. En este artículo, se comenta que el algoritmo tratado en este capítulo no solo es uno de los primeros métodos propuestos para la solución del CVRP, sino que probablemente sea el más utilizado para elaborar la ruta de paquetes a nivel comercial. Como recordatorio, si consideramos la distancia entre un punto i y un punto j como $d_{i,j}$, el máximo ahorro $S_{i,j}$ que se consigue en cada iteración donde se juntan

distintos grupos de puntos es:

$$S_{i,j} = d_{i,0} + d_{0,j} - d_{i,j}. \quad (2.5)$$

En cada iteración, hemos realizado siempre la combinación factible de dos rutas que nos conducen al mayor ahorro posible, y a esto se le denomina como la versión paralela de CW (por Clarke y Wright). Altinel y Oncan [12] afirman que existe otra versión, denominada versión secuencial que se diferencia con la otra a la hora de hacer los emparejamientos. Sin embargo, en el artículo de Laporte y Semet [7] afirman que la versión paralela es mejor que la versión secuencial, basándose en un experimento realizado con unas instancias mostradas por Christofides, Mingozzi y Toth [23].

El ahorro definido en la ecuación 2.5 aumenta cuando la distancia entre el punto i y j es menor en relación con sus distancias al depósito. Pero Altinel y Oncan [12] afirman que a medida que se van haciendo iteraciones los ahorros que se obtienen no son tan grandes como al principio. Para tratar este problema Gaskell [29] y Yellow [25] propusieron la siguiente ecuación de ahorro parametrizada:

$$S_{i,j} = d_{i,0} + d_{0,j} - \lambda d_{i,j}, \quad (2.6)$$

donde λ (solo puede tomar valores positivos) se llama parámetro de forma de ruta y su función es evitar la formación de bucles que afirman que pueden crearse en el algoritmo de Clarke y Wright.

Otra forma de mejorar el rendimiento de este algoritmo propuesta en este artículo es ampliar la condición (2.6) de tal manera que la posición de los puntos influya en el ahorro. La ecuación que se plantea teniendo en cuenta este aspecto es la que propuso Paessens [10], donde se basa en la asimetría entre los puntos i y j respecto a sus distancias al depósito, es decir $d_{i,0} \neq d_{0,j}$. Esta ecuación es la siguiente:

$$S_{i,j} = d_{i,0} + d_{0,j} - \lambda d_{i,j} + \mu |d_{i,0} - d_{0,j}|, \quad (2.7)$$

Aunque los autores no dan detalles sobre la intuición detrás de su idea, parece que lo pretenden primar es que, aunque dos ciudades estén muy distantes entre ellas, el método de solución también las considere como candidatas a reducir la función de costes. Paessens [10] afirma que para encontrar mejores soluciones lo ideal será considerar $0 \leq \lambda \leq 3$ y $0 \leq \mu \leq 1$.

Finalmente, cabe destacar otras dos mejoras de este algoritmo como son las realizadas por Golden, Magnanti y Nguyen [2] y por Nelson, Nygar, Griffin y Shreve [22]. Estos contribuyeron principalmente a la velocidad de cálculo del algoritmo mediante el uso de estructuras de datos especiales para hacer que la determinación del valor de ahorro máximo sea más eficiente computacionalmente.

Capítulo 3

Heurística de ahorro paralelo

A continuación vamos a presentar el trabajo de Altinkemer y Gavish [18] donde presentan un método heurístico de solución del CVRP. Altinkemer y Gavish desarrollan un algoritmo de ahorro paralelo (PSA, siglas que vienen de su traducción en inglés “Parallel Savings Algorithm”) para dicho problema, basado en el artículo de Clarke y Wright que hemos presentado en el Capítulo 2. Altinkemer y Gavish introducen un cambio, que consiste en reemplazar el procedimiento de unión de puntos estacionarios y agregaciones de forma secuencial por un procedimiento basado en un emparejamiento que combina múltiples soluciones parciales en cada paso.

El capítulo está organizado en tres partes. En la Sección 3.1 vamos a mostrar lo que los autores llaman el problema de la entrega y su comparación con el artículo de Clarke y Wright que hemos presentado en el Capítulo 2. En la Sección 3.2, desarrollaremos la heurística de ahorro paralelo para el problema de entrega y mostraremos el algoritmo para resolverlo junto a una serie de resultados matemáticos sobre su orden de complejidad algorítmico. Posteriormente, se aplicará dicho algoritmo a un ejemplo numérico, siguiendo el presentado por Altinkemer y Gavish en su artículo. En la Sección 3.3 explicaremos el concepto de los nodos fantasma, los cuales son explicados a lo largo de todo el artículo original pero que nosotros hemos preferido extraerlos, ya que es algo basado en la intuición de los autores que no es usado para obtener ningún resultado matemático ni en el ejemplo que proponen. Para finalizar, en el Apéndice C, se presenta el problema de emparejamiento de peso máximo, el cual es utilizado en el algoritmo que proponen Altinkemer y Gavish.

3.1. El problema de la entrega

En este artículo nombra al CVRP como “el problema de la entrega” y lo define de la siguiente forma:

“El problema de entrega consiste en encontrar un conjunto de rutas para un grupo de vehículos con capacidades conocidas para satisfacer los requisitos deterministas de entrega de carga de una serie de clientes, los cuales son considerados puntos estacionarios. Los vehículos están ubicados en un depósito central, el terminal, y deben cumplir los requisitos de entrega en una secuencia que minimice los costes totales de entrega. Cada recorrido del vehículo comienza y termina en el terminal, y la carga que se debe entregar a cada punto estacionario es suministrada por un solo vehículo. Un conjunto de puntos estacionarios que está en el mismo recorrido (es decir, servido por el mismo vehículo) forma un grupo.”

Además de esto, consideran la siguiente notación matemática para todos los elementos que aparecen en el problema, la cual hemos cambiado para mantener la misma notación que hemos usado en capítulos anteriores:

$n \equiv$ número de puntos estacionarios, contando el terminal, que aparecen en el problema.

$V \equiv$ conjunto de todos los puntos estacionarios contando al terminal¹, se representa como $V = \{P_1, \dots, P_n\}$.

$\tilde{V} \equiv$ conjunto de todos los puntos estacionarios sin contar al terminal, se representa como $\tilde{V} = \{P_2, \dots, P_n\}$.

$E \equiv$ conjunto de relaciones entre puntos estacionarios, cada una de estas relaciones se escriben $r_{i,j} \forall i, j \in V$.

$C \equiv$ capacidad del vehículo.

$Q_i \equiv$ carga a entregar al punto estacionario P_i ; por completitud $Q_1 = 0$.

$d_{i,j} \equiv$ coste de ir desde el punto P_i al punto P_j ; por completitud consideramos² $d_{i,i} = 0, \forall P_i \in V$.

$x_{i,j} = \begin{cases} 1, & \text{si } r_{i,j} \text{ es recorrida por un vehículo en la solución,} \\ 0, & \text{en caso contrario.} \end{cases}$

$L_S \equiv$ solución óptima para el problema de empaquetamiento de contenedores con S el conjunto total de contenedores mostrada por Eilon y Christofides [28]. Está relacionado con la asignación de los vehículos a los distintos puntos estacionarios a partir de sus respectivas demandas.

Y una vez definidos todos los elementos define el problema como la búsqueda de las variables $x_{i,j}$ tales que minimicen el valor de:

$$D = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{i,j} x_{i,j} + \sum_{i=2}^n d_{i,1} x_{i,1},$$

sujeto a las restricciones:

$$x_{j,1} + \sum_{i=1}^{j-1} x_{i,j} + \sum_{i=j+1}^n x_{j,i} = 2, \quad j = 2, \dots, n-1, \quad (3.1)$$

¹La notación del punto terminal es la única que vamos a modificar respecto a nuestra notación a lo largo de la memoria, denotándolo como P_1 .

²Altinkemer y Gavish definen $d_{i,i} = \infty$; lo hemos modificado ya que es un dato que no se usa, y, además, siguiendo la definición de los autores, infringiría entre otras la desigualdad triangular, la cual asumen los autores para demostrar que la función de ahorro es no negativa.

$$x_{n,1} + \sum_{j=1}^{n-1} x_{j,n} = 2, \quad (3.2)$$

$$\sum_{j=2}^n x_{j,1} \geq L_{\tilde{V}}, \quad (3.3)$$

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n x_{i,j} = n - 1, \quad (3.4)$$

$$\sum_{i \in S} \sum_{i < j \in S} x_{i,j} \leq |S| - L_S, \quad \forall S \subseteq \{2, 3, \dots, n\}, \quad (3.5)$$

$$x_{i,j} \in \{0, 1\}, \quad \forall i, j \in \{1, 2, \dots, n\}. \quad (3.6)$$

Esta formulación es una modificación del problema del vendedor ambulante múltiple que desarrollaron Gavish y Srikanth [1] en 1987. Las restricciones 3.1 y 3.2 aseguran que exactamente un vehículo llega y sale de cada punto estacionario. En la restricción 3.3 se establece que un número idéntico de vehículos debe entrar y salir del terminal, además proporciona un límite inferior en el número de rutas con retorno al terminal. La restricción 3.4 implica que cualquier solución viable debe tener $n - 1$ relaciones entre puntos estacionarios, además de que dichas relaciones acaban regresando directamente al terminal. La restricción 3.5 garantiza que no se puedan formar rutas sin pasar por el terminal y sin violar la restricción de capacidad. Finalmente la restricción 3.6 es conocida como la restricción de integridad.

Para motivar la relevancia del algoritmo que presentan, Altinkemer y Gavish hacen un estudio sobre la complejidad del mismo. Los autores van a presentar tres algoritmos diferentes, que denotan como PSA1, PSA2 y PSA3, de los cuales PSA1 tiene un orden de complejidad superior al polinomial, PSA2 tiene un orden de complejidad de $O(n^3)$ y PSA3 es una leve mejora de PSA2.

3.2. Heurística de ahorro paralelo

Las heurísticas de ahorro paralelo son heurísticas basadas en emparejamientos. Se introdujeron por primera vez y se aplicaron a los sistemas de árboles capacitados por Gavish y Altinkemer [17]. Estas heurísticas de ahorro paralelo combinan múltiples agrupaciones de nodos en cada iteración del algoritmo y el número de agrupaciones fusionadas en cada iteración se determina resolviendo un problema de emparejamiento, maximizando los ahorros obtenidos en la presente iteración. En el caso de Clarke y Wright y de otros trabajos como el de Golden [2] en 1977 y el de Ong y Moore [11] en 1984 también hacían múltiples agrupaciones de nodos, sin embargo, en cada paso solo se fusionan dos agrupaciones de nodos, como hemos visto en el capítulo anterior.

En esta sección, se presenta un PSA para el problema de entrega. La matriz de costos del problema es simétrica con $d_{i,i} = 0$, como hemos definido en

la Sección 3.1. Además todos los costos son no negativos y satisfacen la desigualdad triangular ($d_{k,l} \leq d_{k,1} + d_{1,l}$) y se denomina como Q_i a la carga que se entregará en el punto estacionario i . También hay que destacar la notación utilizada, que es la siguiente:

$R_p \equiv$ conjunto de puntos estacionarios en el conjunto p , el cual es el conjunto de puntos asignados al vehículo p .

$K(R_p) \equiv$ coste de la visita óptima del vendedor viajero entre R_p y el punto estacionario 1, el terminal. Este coste en el PSA1 se calcula resolviendo un TSP.

$S_{p,q} \equiv$ ahorro obtenidos al juntar los grupos p y q .

$\tilde{V}_i \equiv$ conjunto de grupos al final de la iteración i sin incluir el terminal.

$V_i \equiv$ conjunto de grupos al final de la iteración i incluyendo el terminal.

$p_F \equiv$ primer punto del recorrido en el grupo R_p .

$p_L \equiv$ último punto del recorrido en el grupo R_p .

El PSA se aplica al grafo original, con V como su conjunto de nodos o puntos estacionarios y E como su conjunto de aristas o relaciones entre los puntos. En la primera iteración, cada grupo consta de un solo punto estacionario del grafo original. Cuando dos grupos se juntan, los puntos de esos grupos son atendidos por un solo vehículo. La ruta óptima una vez que se tiene un conjunto de puntos se obtiene resolviendo un TSP a la unión de los puntos del conjunto y al terminal. Una vez que se hace una unión y no viola la restricción de capacidad se calcula el valor del ahorro $S_{p,q} = K(R_p) + K(R_q) - K(R_p \cup R_q)$.

Después se resuelve el problema de emparejamiento de peso máximo (ver Apéndice C) en el grafo $G(V, E, S)$, donde S se usa como entradas de costo. Los nuevos grupos se van formando a partir de los grupos previamente formados y este procedimiento iterativo se repite hasta que todas las agrupaciones restantes no sean admisibles. Una declaración formal del algoritmo se presenta a continuación.

- **Paso 0 (Inicialización):** Para $p = 1, \dots, n$, se establece $R_p = \{P_p\}$, $i = 1$ y $V_0 = V$.
- **Paso 1:** Siempre que $p < q$ con $P_p, P_q \in V_{i-1}$:
 - a) Si $\sum_{i \in R_p \cup R_q} Q_i > C$ se pasa a la siguiente combinación de p y q , ya que esa combinación no es admisible al no poderse recorrer con un solo vehículo.
 - b) De lo contrario, se calcula $S_{p,q}$ de acuerdo con la fórmula 3.7.

$$S_{p,q} = \begin{cases} 0, & \text{si } p = 1, q = 1, \dots, |V_{i-1}|, \\ B_{p,q}, & \text{en caso contrario,} \end{cases} \quad (3.7)$$

siendo $B_{p,q} = K(R_p) + K(R_q) - K(R_p \cup R_q)$ y $S_{p,q}$ el valor de los ahorros exactos cuando se juntan P_p y P_q .

- **Paso 2:** Se resuelve el problema de emparejamiento de peso máximo con V_{i-1} como su conjunto de puntos, y $S_{p,q}$ como el coste para $r_{p,q}$. En el caso de que $r_{p,q}$ sea una de las relaciones seleccionadas tras realizar el problema de emparejamiento de peso máximo y $R_p, R_q \in V_{i-1}$ se combinan

los grupos R_p y R_q , a menos que R_p o R_q corresponda con el punto estacionario P_1 , es decir el terminal.

- **Paso 3:** Si todas las posibles agrupaciones son no admisibles hay que parar. Si quedan agrupaciones admisibles, se aumenta i en uno, se actualiza V_i para incluir todos los grupos y se vuelve al Paso 1.

En PSA2 y PSA3, en lugar de resolver un TSP para calcular los ahorros en cada iteración, se calcula una estimación de los ahorros generados por la combinación de dos grupos. La fórmula de ahorro utilizada en PSA2 y PSA3 la siguiente:

$$S_{p,q} = \max \begin{cases} d_{1,p_F} + d_{1,q_F} - d_{p_F,q_F}, \\ d_{1,p_F} + d_{1,q_L} - d_{p_F,q_L}, \\ d_{1,p_L} + d_{1,q_L} - d_{p_L,q_L}, \\ d_{1,p_L} + d_{1,q_F} - d_{p_L,q_F}. \end{cases} \quad (3.8)$$

La fórmula asume que el recorrido en el grupo combinado será la combinación de los recorridos en los grupos R_p y R_q unidos por los puntos finales de los rutas en esos grupos, subestimando así los posibles ahorros al fusionar los grupos R_p y R_q . Bajo este supuesto, el orden de los puntos visitados en el recorrido combinado no se modifica, aunque su reorganización podría mejorar la solución. La única razón para usar una estimación en lugar de resolver el TSP es obtener un algoritmo polinómico.

La única diferencia entre PSA2 y PSA3 consiste en que en PSA3, una vez se han obtenido los grupos finales con la fórmula de ahorros 3.8, para cada grupo final se optimiza su recorrido calculando el TSP para los nodos de ese grupo.

A continuación vamos a mostrar una serie de resultados matemáticos sobre el algoritmo presentado que vienen en el artículo original con su respectiva demostración. El primero de ellos muestra que los ahorros $S_{p,q}$ son siempre no negativos:

Lema 1. $S_{p,q} \geq 0, \forall p, q$.

Demostración. Lo primero, suponemos que $R_p = P_j, \dots, P_k$, $R_q = P_l, \dots, P_m$, $K(R_p) = d_{1,j} + \dots + d_{k,1}$ y $K(R_q) = d_{1,l} + \dots + d_{m,1}$. Si la combinación de los grupos R_p y R_q satisface la restricción de capacidad, formamos una ruta en el grupo combinado tomando las dos rutas, eliminando los enlaces (P_k, P_1) y (P_1, P_l) y agregando el enlace (P_k, P_l) a la solución.

Debido a que $K(R_p \cup R_q)$ corresponde al coste del recorrido óptimo del vendedor en el conjunto combinado entre R_p y R_q tenemos que:

$$K(R_p \cup R_q) \leq d_{1,j} + \dots + d_{k,1} + \dots + d_{m,l}.$$

Y debido a la desigualdad triangular tenemos que:

$$K(R_p \cup R_q) \leq d_{1,j} + \dots + d_{k,1} + \dots + d_{m,l} \leq K(R_p) + K(R_q).$$

Por lo tanto $\forall p, q$:

$$S_{p,q} = K(R_p) + K(R_q) - K(R_p \cup R_q) \geq K(R_p \cup R_q) - K(R_p \cup R_q) = 0,$$

es decir, todos los ahorros son siempre no negativos. \square

El Lema 2 y el Corolario 1 establecen un límite superior en el número de iteraciones realizadas por PSA1 teniendo en cuenta que R_p^m es el conjunto de puntos del grupo R_p al final de la iteración m .

Lema 2. *Al final de la iteración m , $\sum_{P_p \in A_m^i} |R_p^m| > 2^{m+1}$ para todos los A_m^i cuando el límite de capacidad C corresponde con el número máximo de puntos que puede tener una ruta. A_m^i es cualquier conjunto de $2^i + 1$ grupos distintos al final de la iteración m para $i = 0, 1, \dots, \lfloor \log_2 \tilde{V}_m - 1 \rfloor$ tal que $A_m^i \subset \tilde{V}_m$, $|A_m^i| = 2^i + 1$ y $|A_m^i| \leq \tilde{V}_m$.*

Demostración. Vamos a probarlo por inducción sobre el número de iteraciones. Primero notamos que el problema de emparejamiento de peso máximo con entradas de costos no negativos genera un emparejamiento de cardinalidad máxima como solución. Esto es trivial ya que con el problema de emparejamiento de peso máximo siempre se obtiene el mayor coste posible y un emparejamiento de cardinalidad máxima se obtiene cuando el máximo número de vértices posibles es seleccionado en la solución (en caso de tener un número par de vértices se seleccionan todos los vértices y en caso de tener un número impar se seleccionan todos menos uno). Así que si no se seleccionan todos los vértices posibles, la solución que se obtiene no sería la de mayor coste posible ya que todas las entradas de costo son no negativas, llegando así a una contradicción³.

Debido al Lema 1, en cada iteración se pueden emparejar dos vértices no emparejados sin disminuir el coste de la solución. Cuando $|\tilde{V}_0|$ es par, entonces todos los grupos al final de la primera iteración tienen exactamente dos nodos. En ese caso

$$\sum_{P_p \in A_1^i} |R_p^1| = 2 \cdot (2^i + 1) = 2^{i+1} + 2 > 2^{i+1}$$

para todos los A_1^i que satisfacen las condiciones del lema. Si $|\tilde{V}_0|$ es impar, un grupo consta de un punto estacionario y todos los demás grupos tienen exactamente dos puntos. Para A_1^i , que consiste en grupos que solo tienen dos puntos, la igualdad anterior se mantiene claramente. En otros casos, algunos A_1^i contienen el grupo que tiene un punto. En particular para este caso tenemos:

$$\sum_{P_p \in A_1^i} |R_p^1| \geq 2^{1+i} + 1 > 2^{1+i}.$$

³Puede existir el caso de que haya entradas de costo igual a 0 y por eso no se consigue emparejamiento de cardinalidad máxima, sin embargo para conseguirlo se añaden a la solución ya que el coste máximo sigue siendo el mismo.

Por lo tanto, el lema es correcto para la primera iteración. Ahora, supongamos que el lema también es verdadero al final de la iteración k , es decir

$$\sum_{P_p \in A_k^i} |R_p^k| > 2^{k+i}$$

para todos los A_k^i que satisfacen las condiciones del lema. Por el Lema 1 los ahorros son no negativos, dependiendo del número de agrupaciones al final de la iteración k , puede haber como máximo un grupo que quede sin emparejar en la iteración $k+1$. Por lo tanto, para cualquier A_{k+1}^i al final de la iteración $k+1$, hay \tilde{A}_k^{i+1} conjuntos de grupos de la iteración anterior, fusionados para formar las agrupaciones en A_{k+1}^i . En algunos casos, A_{k+1}^i puede incluir un conjunto de puntos en un grupo distinto del conjunto de puntos incluido en el conjunto de grupos en \tilde{A}_k^{i+1} . En esos casos, ese grupo adicional se excluye de \tilde{A}_k^{i+1} . Como resultado, tenemos:

$$\sum_{P_p \in A_{k+1}^i} |R_p^{k+1}| \geq \sum_{P_q \in \tilde{A}_k^{i+1}} |R_q^k| > 2^{k+i+1}.$$

Por lo tanto, el lema se cumple para la iteración $k+1$. \square

Corolario 1. *PSA1 termina después de $\lceil \log_2 C \rceil$ iteraciones cuando se da la restricción de capacidad C como el número máximo de puntos estacionarios que puede tener un recorrido.*

Demostración. Del Lema 2 se deduce:

$$\lfloor R_p^{\lceil \log_2 C \rceil} \rfloor + \lfloor R_q^{\lceil \log_2 C \rceil} \rfloor > 2^{\lceil \log_2 C \rceil}$$

para todos los pares de p y q tales que $P_p, P_q \in \tilde{V}_{\lceil \log_2 C \rceil}$ al final de la iteración $\lceil \log_2 C \rceil$. Si $\lfloor \log_2 C \rfloor = \lceil \log_2 C \rceil$, las fusiones adicionales no son posibles porque violarán la restricción de capacidad y el algoritmo terminará⁴.

Nuevamente del Lema 2 tenemos:

$$\lfloor R_p^{\lceil \log_2 C \rceil} \rfloor + \lfloor R_q^{\lceil \log_2 C \rceil} \rfloor + \lfloor R_s^{\lceil \log_2 C \rceil} \rfloor > 2^{\lceil \log_2 C \rceil + 1}$$

para todos los tríos p, q y s tales que $P_p, P_q, P_s \in \tilde{V}_{\lceil \log_2 C \rceil}$ al final de la iteración $\lceil \log_2 C \rceil$. Si $\lfloor \log_2 C \rfloor \neq \lceil \log_2 C \rceil$ todos los pares de grupos que no violen la restricción de capacidad se fusionarán en la iteración $\lceil \log_2 C \rceil$, excepto como mucho un grupo que no será emparejado. Cualquiera de los dos pares de agrupaciones formadas en la iteración $\lceil \log_2 C \rceil$ contienen al menos tres agrupaciones de la iteración $\lceil \log_2 C \rceil$. De esta desigualdad se deduce que cualquier fusión adicional violaría la restricción de capacidad. Por lo tanto, el algoritmo termina después de $\lceil \log_2 C \rceil$ iteraciones.

En el caso de que los puntos estacionarios tengan cargas desiguales Altinkemer y Gavish afirman que el número máximo de iteraciones está limitado por $\lceil \log_2 C \rceil$. Este es un razonamiento similar al mostrado en el Capítulo 1, cuando

⁴Se definen $\lfloor a \rfloor$ y $\lceil a \rceil$ como la parte entera de a aproximada por defecto y por exceso respectivamente, $\forall a \in \mathbb{R}$.

Dantzig y Ramser decidían el número de etapas de agregación a realizar. Allí, al igual que aquí, se elegía el máximo número de iteraciones a partir del número máximo de nodos cuya demanda pueda cubrir un vehículo, que en este caso son $\lceil \log_2 C \rceil$ iteraciones. \square

PSA1 encuentra la solución óptima para el problema de entrega cuando las agrupaciones pueden contener como máximo dos puntos además del terminal. El siguiente lema aclara este caso especial.

Lema 3. *PSA1 encuentra el resultado óptimo para el problema de entrega cuando las capacidades se dan con un máximo de dos puntos estacionarios por vehículo. Si $H_1(C)$ es la solución PSA1 con una restricción de capacidad de C puntos por vehículo y $Z(C)$ es la solución óptima correspondiente, entonces $H_1(2) = Z(2)$.*

Demostración. Las relaciones en la solución óptima pueden dividirse en dos grupos: el primero consiste en las relaciones que conectan los puntos estacionarios al terminal, y el segundo grupo son las relaciones que interconectan puntos entre sí. El segundo grupo de relaciones son exactamente las relaciones que corresponden al emparejamiento de costos mínimo en el grafo original. Por lo tanto, la inclusión de esas relaciones logra el máximo ahorro en la primera iteración de la primera PSA, ya que al ser la primera iteración se han calculados todos los posibles ahorros, pudiendo seleccionar así el óptimo. En el caso de que n sea impar⁵ no hay ambigüedad. Si n es par, se agrega un nodo imaginario $(n+1)$ tal que $d_{1,n+1} = 0$ y $d_{i,n+1} = d_{i,1}$ para $i = 2, \dots, n$, es decir corresponde con un nodo situado en la misma ubicación que el terminal añadido para poder juntar todos puntos estacionarios en parejas, siendo así un caso similar al de n impar. Este nodo imaginario en la solución final se sustituye por el terminal y el resto de agrupaciones se unen con el terminal tanto n sea par como sea impar, consiguiendo así la solución óptima para $C = 2$. \square

Ejemplo numérico

Para ilustrar el algoritmo de ahorro paralelo, lo aplicamos a un problema que consta de diez ubicaciones, teniendo en cuenta que en cada punto estacionario debe ser entregada una sola unidad de carga, es decir que para todo $j \in \{1, \dots, 10\}$ tenemos que $Q_j = 1$. El terminal está en la ubicación uno y la capacidad del vehículo es de cuatro unidades ($C = 4$). La matriz de distancia $d_{i,j}$ para el ejemplo se muestra en la Figura 3.1. Inicialmente, cada grupo tiene relacionado un único punto estacionario, como se puede ver en la Figura 3.2, por lo que cada nodo es servido por un vehículo separado, así que $R_p = \{P_p\}$ y $K(R_p) = 2d_{1,p}$. Dado que cada par satisface la restricción de capacidad, $S_{p,q}$ se calcula para cada par de grupos teniendo en cuenta que $S_{p,q} = S_{q,p}$. Para la primera iteración tenemos en cuenta que:

$$K(R_p \cup R_q) = d_{p,q} + d_{1,p} + d_{1,q},$$

así que los ahorros $S_{p,q}$ están dados por:

$$S_{p,q} = K(R_p) + K(R_q) - K(R_p \cup R_q) =$$

⁵Si n es impar entonces $n-1$ es par, por lo que se pueden hacer exactamente $\frac{n-1}{2}$ parejas con puntos distintos al terminal.

P1										
164	P2									
170	64	P3								
104	261	274	P4							
181	344	350	89	P5						
243	400	413	140	74	P6					
129	281	299	27	82	120	P7				
169	323	339	65	59	79	41	P8			
203	366	369	113	25	70	106	80	P9		
184	124	61	286	351	419	312	350	368	P10	

Figura 3.1: Matriz de distancias al inicio del ejemplo.

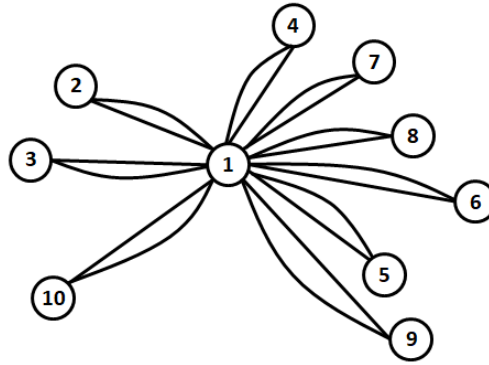


Figura 3.2: Representación gráfica de los puntos al inicio del ejemplo.

$$= 2d_{1,p} + 2d_{1,q} - (d_{p,q} + d_{1,p} + d_{1,q}) = d_{1,p} + d_{1,q} - d_{p,q}.$$

Después se resuelve el problema de emparejamiento de peso máximo con la matriz de costos que se muestra en la Figura 3.3. Como resultado, se forman las siguientes agrupaciones que se pueden ver en la Figura 3.4, las cuales producen los ahorros máximos en la solución inicial: $R_1 = \{P_1\}$, $R_2 = \{P_2\}$, $R_3 = \{P_3, P_{10}\}$, $R_4 = \{P_4, P_7\}$, $R_5 = \{P_5, P_9\}$ y $R_6 = \{P_6, P_8\}$. Cabe destacar que en la Figura 3.4 hemos corregido la imagen que venía en el artículo original, donde se hace una agrupación de cuatro nodos que no tiene sentido en esta primera iteración.

Dado que la carga total en cada grupo es a lo sumo dos es factible fusio-

R1										
0	R2									
0	270	R3								
0	7	0	R4							
0	1	1	196	R5						
0	7	0	207	350	R6					
0	12	0	206	228	252	R7				
0	10	0	208	291	333	257	R8			
0	1	4	194	359	376	226	292	R9		
0	224	293	2	14	8	1	3	19	R10	

Figura 3.3: Matriz de costos tras la primera iteración

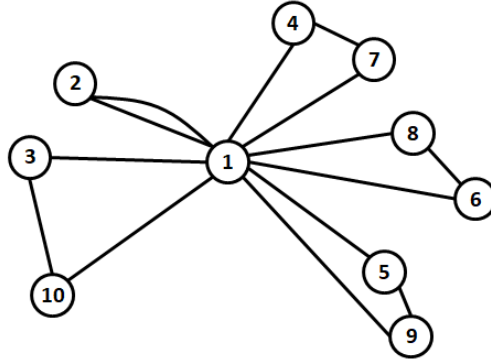


Figura 3.4: Representación gráfica de los puntos tras la primera iteración.

R1					
0	R2				
0	270	R3			
0	12	2	R4		
0	1	19	228	R5	
0	10	8	258	376	R6

Figura 3.5: Matriz de costos tras la segunda iteración

narlas, así que calculamos $S_{p,q}$ para cada par de grupos:

$$S_{p,q} = \begin{cases} 0, & \text{si } p = 1, q = 1, \dots, |V_{i-1}|, \\ B_{p,q}, & \text{en caso contrario,} \end{cases}$$

siendo $B_{p,q} = K(R_p) + K(R_q) - K(R_p \cup R_q)$.

En este paso, los ahorros $S_{p,q}$ se calculan a partir del costo óptimo de las rutas en $R_p \cup \{P_1\}$ y $R_q \cup \{P_1\}$ tras realizar un TSP. Posteriormente se hace lo mismo con el costo de la ruta en el grupo $R_p \cup R_q \cup \{P_1\}$, al cual se le resta la suma de los dos primeros costos obteniendo así el valor de cada $S_{p,q}$.

El problema de emparejamiento de peso máximo se resuelve con la nueva matriz de costos como se muestra en la Figura 3.5, lo que lleva a las agrupaciones que se ven en la Figura 3.6: $R_1 = \{P_1\}$, $R_2 = \{P_2, P_3, P_{10}\}$, $R_3 = \{P_4, P_7\}$ y $R_4 = \{P_5, P_6, P_8, P_9\}$. Esta es la iteración final del algoritmo con un costo de 1,257, ya que al juntar cualquiera dos grupos se violaría la restricción de capacidad, debido que se sobrepasaría la carga máxima de cuatro unidades descrita al principio del ejemplo.

3.3. Ampliación: uso de nodos fantasma

A lo largo de todo el artículo de Altinkemer y Gavish [18] se hace referencia a unos nodos denominados nodos fantasma, que se introducen durante las

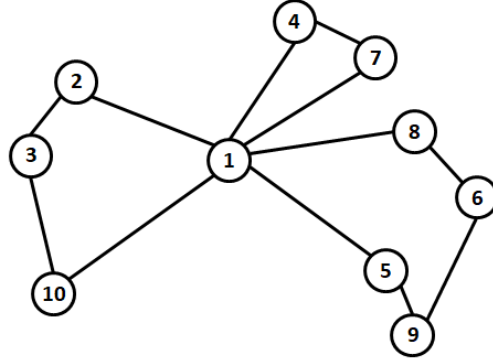


Figura 3.6: Representación gráfica de los puntos tras la segunda iteración

iteraciones con el propósito de reducir la velocidad de solución para conseguir mejores soluciones.

Los autores del artículo introducen una heurística la cual está basada en su intuición y crean un conjunto⁶ de nodos fantasma al que denotan I_i , siendo i el número de la iteración en la que se encuentran. Además toman el siguiente valor: $|I_i| = n - (i + 1)T$, siendo el valor T un número entero tal que $2 \leq T \leq 12$.

Este conjunto de nodos fantasma se introduce en el Paso 1 del algoritmo descrito en la Sección 3.2, donde los ahorros de sus nodos se consideran de la siguiente forma⁷:

$$S_{p,q} = \begin{cases} 0, & \text{si } p < q \text{ y } P_p, P_q \in I_i. \\ +\infty, & \text{si } p < q \text{ y } P_p \in V_{i-1} \text{ y } P_q \in I_i. \end{cases}$$

Posteriormente, en el Paso 2, se resuelve el problema de emparejamiento de peso máximo con el conjunto $V_{i-1} \cup I_i$ en vez de como habíamos indicado anteriormente con el conjunto V_{i-1} . A partir de ese punto se resuelve de la misma forma con el único cambio de que habrá más iteraciones hasta llegar a la solución final.

En resumen, se podría decir que la idea clave de ese valor T es marcar el número de agrupaciones que se quiere forzar a realizar en cada iteración intentando ralentizar la velocidad del algoritmo. Esto se hace ya que el algoritmo mostrado es un “algoritmo voraz”, el cual una vez realizadas las agrupaciones ya no se pueden deshacer y si se van haciendo menos agrupaciones por iteración se pueden conseguir más posibles agrupaciones, pudiendo llegar así a una solución mejor. Para que quede más claro el por qué T es el número de agrupaciones por iteración vamos a mostrar qué pasaría en la primera de ellas:

⁶En el artículo original se denota M_i al conjunto de los nodos fantasma, sin embargo hemos optado por denominarla I_i para no confundirla con la matriz M utilizada en otros capítulos.

⁷En el artículo original pone que el ahorro entre un nodo fantasma y un punto estacionario es igual a $-\infty$, sin embargo al resolver el problema de emparejamiento de peso máximo eso no tendría sentido ya que esos nunca serían escogidos y la idea de estos nodos es forzar a que se den esas agrupaciones.

Problema		Valor de la T									Mejor solución conocida en la literatura (hasta 1991)
Nº	Tamaño (nodos)	2	3	4	6	7	9	10	11	12	
1	51	574	560	582	571	574	577	585	580	568	524
2	76	857	866	894	889	883	975	875	895	870	857
3	101	870	903	892	881	861	874	892	872	877	833
4	151	1112	1117	1127	1111	1085	1089	1102	1085	1087	1014
5	200	1357	1357	1364	1351	1396	1368	1388	1357	1367	1386
6	51	620	620	595	599	583	598	604	613	624	560
7	76	943	954	969	977	966	955	977	1016	988	916
8	101	945	974	982	956	919	913	937	916	936	885
9	151	1270	1320	1274	1258	1299	1251	1244	1211	1260	1217
10	200	1571	1573	1534	1562	1529	1489	1514	1494	1464	1508
11	121	1052	1047	1089	1097	1130	1052	1047	1089	1054	1079
12	101	834	834	834	839	842	849	866	855	850	816
13	121	1559	1551	1560	1579	1625	1644	1607	1642	1622	1590
14	101	874	875	874	874	929	889	886	881	971	848

Figura 3.7: Tabla de soluciones para diferentes valores de T .

- Primero tenemos que $|V_0| = n$ y que $i = 1$.
- Añadimos los nodos fantasma con $|I_1| = n - (i + 1)T = n - 2T$ y definimos los ahorros como se ha explicado previamente.
- Resolvemos el problema de emparejamiento de peso máximo sobre el conjunto $V_0 \cup I_1$, teniendo en cuenta que $|V_0 \cup I_1| = n + (n - 2T) = 2n - 2T$.
- Al hacer los emparejamientos, se genera un conjunto maximal de emparejamientos, con $n - T$ parejas.
- Todos los nodos de I_i se han emparejado con algún nodo de V_0 , porque su ahorro es $+\infty$.
- Por tanto, tenemos $n - T$ emparejamientos, de los cuales $n - 2T$ son de los puntos de I_1 con puntos de V_0 (recordamos que $|I_1| = n - 2T$).
- Como $n - T = (n - 2T) + T$, tenemos que de todos los nuevos emparejamientos, $n - 2T$ son entre nodos fantasma, los cuales los descartamos, y T son entre nodos de V_0 , y por tanto son aquellos los que se meten en la solución final.

Para finalizar este apartado queremos mostrar una tabla de resultados (ver Figura 3.7 donde las casillas que están en verde son aquellas que consiguen el mejor resultado para cada problema) que los autores tienen en su artículo, en la cual resuelven catorce problemas diferentes usando los nodos fantasma dándole todos los valores posibles a T (menos $T = 5$ y $T = 8$, ya que para esos dos valores no se consigue la mejor solución en ninguno de los catorce problemas) y viendo que dependiendo del problema se consigue la mejor solución con un diferente valor de T sin ninguna justificación por parte de los autores, más allá de los resultados empíricos.

Conclusiones y Trabajo Futuro

Tras realizar el trabajo fin de grado me quedo con un buen sabor de boca debido a que después de muchas semanas trabajando en entender todos los cálculos que hicieron los autores de estos tres artículos, hemos conseguido realizarlos y llegar a las soluciones que se nos muestran. El artículo que más trabajo nos ha dado respecto a este aspecto es el mostrado en el Capítulo 1, el “Problema del reparto de camiones” de Dantzig y Ramser, debido a que había que realizar muchas iteraciones en las cuales aparecían complicaciones que no venían bien descritas en el artículo original.

Sin embargo, tras solventar esas complicaciones realizando todos los cálculos (ver Apéndice A) hemos conseguido entender todo lo que nos mostraban los autores en ese artículo y entender las dificultades que pudieron tener ellos en el año 1959 ya que no contaban con las herramientas que tenemos ahora. Otro aspecto importante que quiero destacar de ese artículo es la importancia que ha tenido en la historia, ya que fue el artículo donde se dio la primera formulación del CVRP y es el que ha inspirado a todos los demás autores a intentar mejorar tanto de forma aproximada como exacta el algoritmo dado en dicho artículo.

Por el contrario, respecto al Capítulo 2 quiero destacar lo bien que estaba explicado el artículo original ya que nos ha resultado mucho más sencillo entender todo lo que se hacía y llegar a la solución que se daba, la cual, a pesar de la simplicidad del algoritmo, mejora la solución dada en el artículo anterior. De hecho, los autores afirman que llegaron a conseguir la que Dantzig y Ramser consideraban la solución óptima del ejemplo que hemos mostrado en ambos capítulos.

Respecto al Capítulo 3, quiero destacar que en la mayoría de los artículos posteriores que hemos leído, aunque el artículo aparece citado como una referencia relevante en el campo (posiblemente por la novedosa aportación de calcular los ahorros “en paralelo”), en ninguno de los artículos posteriores se llega a detallar el método ni se nombra a los nodos fantasma. Eso nos ha hecho hacer alguna suposición sobre este aspecto y separar el algoritmo en dos, lo cual hace que el capítulo en mi opinión quede mejor organizado y que sea mucho más fácil de leer y entender.

Finalmente, definiría mi trabajo como un inicio de un proyecto en el que

poder seguir analizando diferentes algoritmos del CVRP, ya que este sería como una primera parte al habernos centrado en los primeros algoritmos que consiguen soluciones aproximadas, pudiendo hacer una continuación centrándonos en algoritmos que consiguen soluciones exactas. Además, para finalizar, quiero destacar que este trabajo me ha servido para mejorar mi forma de buscar información en artículos principalmente matemáticos y que la única “espinita” que se me ha podido quedar clavada es no haber implementado el algoritmo del Capítulo 1, del cual no hemos podido encontrar ninguna implementación, ya que nos hubiera llevado a realizar un proyecto solamente centrado en él debido a la cantidad de suposiciones y cálculos que hay que hacer, no pudiendo analizar los otros dos artículos en los que hemos trabajado.

Bibliografía

- [1] B. GAVISH Y K. SRIKANTH. An optimal solution method for large-scale multiple traveling salesmen problems. *Operations Research* 34, 5 (1986), 698–717.
- [2] B. GOLDEN, T.L. MAGNANTI Y H.Q. NGUYEN. Implementing vehicle routing algorithms. *Networks* 7, 2 (1977), 113–148.
- [3] E. BENAVENT, M. LANDETE, E. MOTA Y G. TIRADO. The multiple vehicle pickup and delivery problem with LIFO constraints. *European Journal of Operational Research* 243, 3 (2015), 752–762.
- [4] F. LI, B. GOLDEN Y E. WASIL. Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research* 32, 5 (2005), 1165–1179.
- [5] G. CLARKE Y J. WRIGHT. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research* 12, 4 (1964), 568–581.
- [6] G. DANTZIG Y J. RAMSER. The truck dispatching problem. *Management science* 6, 1 (1959), 80–91.
- [7] G. LAPORTE Y F. SEMET. Classical heuristics for the vehicle routing problem. *Cahiers du GERAD* (1999).
- [8] H. MIN. The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General* 23, 5 (1989), 377–386.
- [9] H. N. GABOW. Implementation of algorithms for maximum matching on nonbipartite graphs.
- [10] H. PAESSENS. The savings algorithm for the vehicle routing problem. *European Journal of Operational Research* 34, 3 (1988), 336–344.
- [11] H.L. ONG Y J.B. MOORE. Worst-case analysis of two travelling salesman heuristics. *Operations Research Letters* 2, 6 (1984), 273–277.
- [12] İ.K. ALTINEL Y T. ÖNCAN. A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. *Journal of the Operational Research Society* 56, 8 (2005), 954–961.
- [13] J. EDMONDS. Paths, trees, and flowers. *Canadian Journal of mathematics* 17 (1965), 449–467.

- [14] J.A. TOMPKINS, J.A. WHITE, Y.A. BOZER Y J.M.A. TANCHOCO. *Facilities planning*. John Wiley & Sons, 2010.
- [15] J.K. LENSTRA Y A.R. KAN. Complexity of vehicle routing and scheduling problems. *Networks* 11, 2 (1981), 221–227.
- [16] JORIS VR. <http://jorisvr.nl/article/maximum-matching>, June, 8th 2013.
- [17] K. ALTINKEMER Y B. GAVISH. Heuristics with constant error guarantees for the design of tree networks. *Management Science* 34, 3 (1988), 331–341.
- [18] K. ALTINKEMER Y B. GAVISH. Parallel savings based heuristics for the delivery problem. *Operations Research* 39, 3 (1991), 456–469.
- [19] K.J. ROODBERGEN. Routing order pickers in a warehouse, <http://www.roodbergen.com/warehouse/background.php>.
- [20] M. CARRILLO Y A. FELIPE. Modelo de ruteo de vehículos para la distribución de las empresas laboratorios Veterland, Laboratorios Callbest y Cosméticos Marliou Paris. B.S. Tesis, Facultad de Ingeniería, 2014.
- [21] M. DESROCHERS, J. DESROSIERS Y M. SOLOMON. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research* 40, 2 (1992), 342–354.
- [22] M.D. NELSON, K.E. NYGARD, J.H. GRIFFIN Y W.E. SHREVE. Implementation techniques for the vehicle routing problem. *Computers & Operations Research* 12, 3 (1985), 273–283.
- [23] N. CHRISTOFIDES, A. MINGOZZI, P. TOTH Y C. SANDI. The vehicle routing problem. Combinatorial optimization, 1979.
- [24] P. TOTH Y D. VIGO. *The vehicle routing problem*. SIAM, 2002.
- [25] P.C. YELLOW. A computational modification to the savings method of vehicle scheduling. *Journal of the Operational Research Society* 21, 2 (1970), 281–283.
- [26] P.M. FRANCIS, K.R. SMILOWITZ Y M. TZUR. The period vehicle routing problem and its extensions. In *The vehicle routing problem: latest advances and new challenges*. Springer, 2008, pp. 73–102.
- [27] P.P. BALLESTEROS Y A. ESCOBAR. Revisión del estado del arte del problema de ruteo de vehículos con recogida y entrega (VRPPD). *Ingeniería y Desarrollo* 34, 2 (2016), 463–482.
- [28] S. EILON Y N. CHRISTOFIDES. The Loading problem. *Manage. Sci.* 17, 5 (Jan. 1971), 259–268.
- [29] T.J. GASKELL. Bases for vehicle fleet scheduling. *Journal of the Operational Research Society* 18, 3 (1967), 281–295.
- [30] Z. FU, R. EGGLESE, Y L.Y. LI. A new tabu search heuristic for the open vehicle routing problem. *Journal of the operational Research Society* 56, 3 (2005), 267–274.

- [31] Z. GALIL. Efficient algorithms for finding maximum matching in graphs.
ACM Computing Surveys (CSUR) 18, 1 (1986), 23–38.

Apéndice A

Ejemplo numérico de Dantzig y Ramser

A.1. Primera etapa de agregaciones

Lo primero de todo, suponemos la solución inicial (ver Figura A.1) emparejando el terminal P_0 con los demás puntos estacionarios (P_1, P_2, \dots, P_{12}), siendo ese el primer conjunto básico (casillas en color verde, mientras que las demás corresponden al conjunto no básico incluidas aquellas en color rojo que corresponde a las entradas que superan los 3000, carga correspondiente a la mitad de la capacidad total).

A.1.1. Correcciones rápidas

El siguiente paso es hacer las correcciones rápidas:

- **Corrección 1:** (ver Figura A.2) como $d_{1,2} = 5$ y $d_{0,1} = 9 < d_{0,2} = 14$:
 - $x_{1,2} = 1$ reemplaza $x_{0,2}$.
 - $x_{0,1} = 1$ pasa a ser $x_{0,1} = 0$.

Q	P0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				</	
---	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----	--

Figura A.1: Tabla del ejemplo al inicio.

Figura A.4: Tabla tras la tercera corrección.

Figura A.5: Tabla tras la cuarta corrección.

Q	P0
1200	9
1700	14
1500	21
1400	23
1700	22
1400	25
1200	32
1900	36
1800	38
1600	42
1700	50
1100	52

Figura A.8: Tabla tras darle valor a la primera variable θ .

Q	P0	0												
1200	0	P1	9											
1700	-18	0	P2	-4										
1500	0	18	10	P3	21									
1400	-34	-24	-32	0	P4	-11								
1700	0	10	2	22	-8	P5	22							
1400	-50	-40	-52	-34	-64	-12	P6	-25						
1200	0	10	2	26	-4	44	0	P7	32					
1900	0	10	2	20	-10	42	0	58	P8	36				
1800	0	10	-2	16	-14	38	0	54	68	P9	38			
1600	0	10	2	32	2	44	0	64	72	68	P10	42		
1700	0	10	2	34	8	44	0	64	72	76	84	P11	50	
1100	-32	-82	-30	-58	-80	-48	-92	-28	-20	-22	-8	0	P12	-40

Figura A.9: Tabla tras el segundo cálculo de la función Delta.

Q	P0
1200	0 ₉
1700	14 ¹
1500	21 ₁₂
1400	23 ₂₂
1700	22 ₂₁
1400	25 ₂₄
1200	32 ₃₁
1900	36 ¹ ₃₅
1800	38 ¹ ₃₇
1600	42 ^{1-2g} ₄₁
1700	50 ₄₉
1100	52 ₅₁

Figura A.10: Tabla al intentar introducir la segunda variable θ .

Figura A.11: Tabla tras intentar darle valor a la segunda variable θ .

Figura A.12: Tabla al introducir la segunda variable θ .

Figura A.13: Tabla tras darle valor a la segunda variable θ .

recalculando todos los π_i y los $\delta_{i,j}$, solo que a partir de ahora las cuentas salen sin complicaciones por lo que las vamos a mostrar de forma resumida:

- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{7,9}$ (ver Figura A.14).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.15).
- A la variable se le da el valor $\theta = 1$ y se elimina la entrada correspondiente al $x_{0,10}$ del conjunto básico (ver Figura A.16).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{3,6}$ (ver Figura A.17).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.18).
- A la variable se le da el valor $\theta = 0$ y se elimina la entrada correspondiente al $x_{0,3}$ del conjunto básico (ver Figura A.19).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{1,6}$ (ver Figura A.20).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.21).
- A la variable se le da el valor $\theta = 0$ y se elimina la entrada correspondiente al $x_{0,9}$ del conjunto básico (ver Figura A.22).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{1,3}$ (ver Figura A.23).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.24).
- A la variable se le da el valor $\theta = 0$ y se elimina la entrada correspondiente al $x_{3,6}$ del conjunto básico (ver Figura A.25).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{1,5}$ (ver Figura A.26).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.27).
- A la variable se le da el valor $\theta = 0$ y se elimina la entrada correspondiente al $x_{0,1}$ del conjunto básico (ver Figura A.28).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{3,8}$, sin embargo si intentamos recalcular los $x_{i,j}$ en función de la variable θ vemos que es imposible. (ver Figura A.29 y A.30).
- Como ya no quedan más $\delta_{i,j} > 0$ se da por concluida la primera etapa de agregaciones.

A.2. Segunda etapa de agregaciones

Como se ha explicado en el Capítulo 1 a partir de la Figura A.28 sacamos las relaciones de la Figura 1.10. A partir de esas relaciones sacamos una tabla similar a la realizada en la primera etapa de agregaciones pero cambiando los P_i por A_i (ver Figura A.31, donde podemos observar como se empieza el algoritmo de la misma forma que en la primera etapa, suponiendo la solución correspondiente a relacionar A_0 con los demás puntos que van desde A_1 hasta A_7).

Q	P0	0													
1200	0	P1	9												
1700	-18	0	P2	-4											
1500	0	18	10	P3	21										
1400	-34	-24	-32	0	P4	-11									
1700	0	10	2	22	-8	P5	22								
1400	-50	-40	-52	-34	-64	-12	P6	-25							
1200	0	10	2	26	-4	44	0	P7	32						
1900	0	10	2	20	-10	42	0	58	P8	36					
1800	0	10	-2	16	-14	38	0	54	68	P9	38				
1600	0	10	2	32	2	44	0	64	72	68	P10	42			
1700	-64	-54	-62	-30	-56	-20	-64	0	8	12	20	P11	-14		
1100	-28	-18	-26	6	-16	16	-28	36	44	42	56	0	P12	24	

Figura A.14: Tabla tras el tercer cálculo de la función Delta.

Q	P0													
1200	0	9	P1											
1700	14	1	5	P2										
1500	0	21	12		7	P3								
1400	23	22	17	1	10	P4								
1700	1	22	21	16	21	19	P5							
1400	25	24	23	30	28	9	P6							
1200	32	31	26	27	25	10	1-e	7	P7					
1900	1	36	35	30	37	35	16	11	10	P8				
1800	1-e	38	37	36	43	41	22	13	6	16	P9			
1600	1-e	42	41	36	31	29	20	0+e	17	10	6	12	P10	
1700	50	49	44	37	31	28	25	0	18	14	12	8		P11
1100	52	51	46	39	29	30	27	20	16	20	10	1	10	P12

Figura A.15: Tabla al introducir la tercera variable θ .

Q	P0																								
1200	0	9	P1																						
1700		14	1	5	P2																				
1500	0	21		12		7	P3																		
1400		23		22		17	1	10	P4																
1700	1	22		21		16		21		19	P5														
1400		25		24		23		30		28		9	P6												
1200		32		31		26		27		25	10	0	7	P7											
1900	1	36		35		30		37		35	16		11		10	P8									
1800	0	38		37		36		43		41	22	13	1	16		6	P9								
1600		42		41		36		31		29	20	1	17		10	6		12	P10						
1700		50		49		44		37		31	28		25	0	18		14	12		8	P11				
1100		52		51		46		39		29	30		27		20	16		20		10	1	10	P12		

Figura A.16: Tabla tras darle valor a la tercera variable θ .

Figura A.17: Tabla tras el cuarto cálculo de la función Delta.

Figura A.18: Tabla al introducir la cuarta variable θ .

Figura A.19: Tabla tras darle valor a la cuarta variable θ .

Figura A.20: Tabla tras el quinto cálculo de la función Delta.

Figura A.21: Tabla al introducir la quinta variable θ .

Figura A.22: Tabla tras darle valor a la quinta variable θ .

Q	P0													
1200	0	P1												
1700	-18	0	P2											
1500	-6	12	4	P3										
1400	-28	-18	-26	0	P4									
1700	0	10	2	16	-2	P5								
1400	-10	0	-12	0	-18	28	P6							
1200	-40	-30	-38	-20	-38	4	0	P7						
1900	0	10	2	14	-4	42	40	18	P8					
1800	-14	-4	-16	-4	-22	24	26	0	54	P9				
1600	-40	-30	-38	-14	-32	4	0	-16	32	14	P10			
1700	-24	-14	-22	4	-10	20	16	0	48	38	20	P11		
1100	-68	-58	-66	-40	-50	-24	-28	-44	4	-12	-24	0	P12	-16

Figura A.23: Tabla tras el sexto cálculo de la función Delta.

Q	P0														
1200	0-8	9	P1												
1700	14	1	5	P2											
1500	21	8	12	7	P3										
1400	23	22	17	1	10	P4									
1700	1	22	21	16	21	19	P5								
1400	25	0	24	23	0-8	30	28	9	P6						
1200	32	31	26	27	25	10	0+8	7	P7						
1900	1	36	35	30	37	35	16	11	10	P8					
1800	38	37	36	43	41	22	13	1	16	6	P9				
1600	42	41	36	31	29	20	1	17	10	6	12	P10			
1700	50	49	44	37	31	28	25	0-8	18	14	12	8	P11		
1100	52	51	46	39	29	30	27	20	16	20	10	1+8	10	P12	

Figura A.24: Tabla al introducir la sexta variable θ .

Q	P0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
---	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Figura A.25: Tabla tras darle valor a la sexta variable θ .

Figura A.26: Tabla tras el séptimo cálculo de la función Delta.

Figura A.27: Tabla al introducir la séptima variable θ .

Figura A.28: Tabla tras darle valor a la séptima variable θ .

Q	P0	0
1200	-10	P1 -1
1700	-8	0 P2 6
1500	-8	0 12 P3 13
1400	-26	-26 -14 0 P4 -3
1700	0	0 12 14 0 P5 22
1400	0	0 8 8 -6 38 P6 25
1200	-50	-50 -38 -32 -46 -6 0 P7 -18
1900	0	0 12 12 -2 42 50 8 P8 36
1800	-4	-4 4 4 -10 34 46 0 64 P9 34
1600	-50	-50 -38 -26 -40 -6 0 -36 22 14 P10 -8
1700	-14	-14 -2 12 2 30 36 0 58 58 20 P11 36
1100	-78	-78 -66 -52 -58 -34 -28 -64 -6 -12 -44 0 P12 -26

Figura A.29: Tabla tras el último cálculo de la función Delta.

[illegible]

Figura A.30: Tabla al intentar introducir la última variable θ .

A0									
1	28	A1							
1	54		A2						
1	44	52		72	A3				
1	86	92	102		84	A4			
1	84	94	110	86		32	A5		
1	72	80	102	74	84		86	A6	
1	112	120	120	112	112	112	112		A7

Figura A.31: Tabla al inicio de la segunda etapa de agregaciones.

El siguiente paso es utilizar la función Delta con sus respectivos valores π_i y $\delta_{i,j}$ con $i \in \{0, 1, 2, \dots, 6\}$ y $j \in \{1, 2, \dots, 7\}$. A partir de ahí es hacer exactamente lo mismo de antes, así que lo mostramos de forma resumida:

- Se calculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{5,7}$ (ver Figura A.32).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.33).
- A la variable se le da el valor $\theta = 1$ y se elimina la entrada correspondiente al $x_{0,7}$ del conjunto básico (ver Figura A.34).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{4,5}$ (ver Figura A.35).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.36).
- A la variable se le da el valor $\theta = 0$ y se elimina la entrada correspondiente al $x_{0,5}$ del conjunto básico (ver Figura A.37).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{4,7}$ (ver Figura A.38). Sin embargo una vez introducida la variable θ , haber realizado los ajustes correspondientes y haberle dado el valor $\theta = \frac{1}{2}$ (ver Figura A.39 y A.40) vemos que cualquier ajuste que se quiera realizar posteriormente en esas entradas lleva a alguna contradicción. Así que seleccionamos el siguiente $\delta_{i,j}$ más grande que es $\delta_{4,6}$.
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.41).
- A la variable se le da el valor $\theta = 1$ y se elimina la entrada correspondiente al $x_{0,4}$ del conjunto básico (ver Figura A.42).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{5,6}$ (ver Figura A.43). Sin embargo después de realizar bastantes operaciones se llega a otra posible solución del problema muy similar a la solución que dan Dantzig y Ramser (ver Figura A.44, A.45, A.46, A.47, A.48, A.49, A.50, A.51, A.52, A.53, A.54 y A.55). Así que decidimos seleccionar el siguiente $\delta_{i,j}$ más grande que es $\delta_{4,6}$ y al final del problema comparar ambas y decir porque optamos por la misma que los autores.
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.56).
- A la variable se le da el valor $\theta = 0$ y se elimina la entrada correspondiente al $x_{0,6}$ del conjunto básico (ver Figura A.57).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{1,2}$ (ver Figura A.58).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.59).
- A la variable se le da el valor $\theta = 1$ y se elimina la entrada correspondiente al $x_{0,2}$ del conjunto básico (ver Figura A.60).
- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{1,3}$ (ver Figura A.61).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.62).
- A la variable se le da el valor $\theta = 0$ y se elimina la entrada correspondiente al $x_{0,1}$ del conjunto básico (ver Figura A.63).

A0	0								
0	A1	28							
0		28	A2	54					
0		20	28	A3	44				
0		20	36		A4	84			
0		20	30	44		78	A5	86	
0		20	24	42	72		A6	72	
0		20	46	44	84	86		A7	112

Figura A.32: Tabla durante la segunda etapa de agregaciones.

A0									
1	A1								
1	54	54	A2						
1	44	52	72	A3					
1	84	92	102		A4				
1-8	86	94	110	86	32	A5			
1	72	80	102	74	84	86	A6		
1-8	112	120	120	112	112	112		A7	

Figura A.33: Tabla durante la segunda etapa de agregaciones.

- Se recalculan los valores y se elige el máximo $\delta_{i,j}$ que es $\delta_{2,3}$ (ver Figura A.64).
- Aparece la variable θ y se ajustan los $x_{i,j}$ (ver Figura A.65).
- A la variable se le da el valor $\theta = \frac{1}{2}$ y se elimina la entrada correspondiente al $x_{0,3}$ del conjunto básico (ver Figura A.66).
- Se recalculan los valores y se ve que no quedan $\delta_{i,j} > 0$, pero se tiene que $\delta_{5,6} = 0$ (ver Figura A.67).

Una vez aquí vemos que hemos llegado a dos soluciones que corresponden a las tablas de las Figura A.54 y A.66, las cuales son similares salvo por una entrada básica que cambia por lo que en este caso al igual que Dantzig y Ramser optamos por elegir la entrada con el correspondiente d_{A_i, A_j} más pequeño y ese es la razón nos quedamos con la relación (A_3, A_6) con $d_{A_3, A_6} = 74$ antes que con la relación (A_5, A_6) con $d_{A_5, A_6} = 86$.

A.2.1. Tratamiento fraccionario de $x_{i,j}$

Si no hubiesen aparecido en la solución de la Figura A.66 valores fraccionarios la segunda etapa de agregaciones, sin embargo al aparecer hay que hacer un tratamiento especial de esas entradas. En esa Figura se muestra que la solución óptima contiene tres entradas básicas fraccionarias que involucran el

A0									
1	28	A1							
1	54	54	A2						
1	44	52	72	A3					
1	84	92	102	84	A4				
0	86	94	110	86		A5			
1	72	80	102	74	84	86	A6		
	112	120	120	112	112	1	112		A7

Figura A.34: Tabla durante la segunda etapa de agregaciones.

A0	0								
0	A1	28							
0	28	A2	54						
0	20	26	A3	44					
0	20	36	44	A4	84				
0	20	30	44	78	A5	86			
0	20	24	42	72	72	A6	72		
-86	-66	-40	-42	-2	0	-14	A7	26	

Figura A.35: Tabla durante la segunda etapa de agregaciones.

A0									
1	28	A1							
1	54	54	A2						
1	44	52	72	A3					
1-8	84	92	102	84	A4				
0-8	86	94	110	86	92	A5			
1	72	80	102	74	84	86	A6		
	112	120	120	112	112	112		A7	

Figura A.36: Tabla durante la segunda etapa de agregaciones.

A0									
1	28	A1							
1	54	54	A2						
1	44	52	72	A3					
1	84	92	102	84	A4				
	86	94	110	86	92	A5			
1	72	80	102	74	84	86	A6		
	112	120	120	112	112	112		A7	

Figura A.37: Tabla durante la segunda etapa de agregaciones.

A0	0								
0	A1	28							
0	28	A2	54						
0	20	26	A3	44					
0	20	36	44	A4	84				
-78	-58	-48	-34	0	A5	8			
0	20	24	42	72	-6	A6	72		
-8	12	38	36	76	0	64	A7	104	

Figura A.38: Tabla durante la segunda etapa de agregaciones.

A0									
1	28	A1							
1	54	54	A2						
1	44	52	72	A3					
1-28	84	92	102	84	A4				
	86	94	110	86	92	A5			
1	72	80	102	74	84	86	A6		
	112	120	120	112	112	112		A7	

Figura A.39: Tabla durante la segunda etapa de agregaciones.

A0	
1	28
1	54
1	44
	84
	86
1	72
112	120

A1	
	54
	52
	92
	94
	80
	120

A2	
	72
	102
	110
	102
	120

A3	
	84
	86
	74
	112

A4	
	32
	112

A5	
	86
	112

A6	
	86
	112

A7	
	112

Figura A.40: Tabla durante la segunda etapa de agregaciones.

A0	
1	28
1	54
1	44
1-5	84
	86
1-5	72
112	120

A1	
	54
	52
	92
	94
	80
	120

A2	
	72
	102
	110
	102
	120

A3	
	84
	86
	74
	112

A4	
	32
	112

A5	
	86
	112

A6	
	86
	112

A7	
	112

Figura A.41: Tabla durante la segunda etapa de agregaciones.

A0	
1	28
1	54
1	44
	84
	86
0	72
112	120

A1	
	54
	52
	92
	94
	80
	120

A2	
	72
	102
	110
	102
	120

A3	
	84
	86
	74
	112

A4	
	32
	112

A5	
	86
	112

A6	
	86
	112

A7	
	112

Figura A.42: Tabla durante la segunda etapa de agregaciones.

A0	
0	28
0	20
-72	-52
-6	14
0	20
-80	-60

A1	
	28
	20
	-52
	24
	24
	-60

A2	
	54
	26
	-36
	24
	24
	-34

A3	
	44
	38
	-28
	42
	-36

A4	
	12
	38
	0
	0
	-68

A5	
	80
	66
	0
	0
	0

A6	
	72
	-8
	-8
	-8
	-8

A7	
	32

Figura A.43: Tabla durante la segunda etapa de agregaciones.

P0	
1	28
1	54
1	44
	84
	86
0-25	72
112	120

P1	
	54
	52
	92
	94
	80
	120

P2	
	72
	102
	110
	102
	120

P3	
	84
	86
	74
	112

P4	
	32
	112

P5	
	86
	112

P6	
	86
	112

P7	
	112

Figura A.44: Tabla durante la segunda etapa de agregaciones.

P0									
1	28	P1							
1	54		54	P2					
1	44		52		72	P3			
	84		92		102		84	P4	
	86		94		110		86	32	P5
	72		80		102		74	84	
	112		120		120		112	112	P6
									P7

Figura A.45: Tabla durante la segunda etapa de agregaciones.

P0	0								
0	P1	28							
0		28	P2	54					
0		20		26	P3	44			
-39		-19		-3		5	P4	45	
-39		-19		-9		5	0	P5	47
-33		-13		-9		9	0		P6
-47		-27		-1		-3	-2	0	P7
									65

Figura A.46: Tabla durante la segunda etapa de agregaciones.

P0									
1-8	28	P1							
1-8	54		54	P2					
1	44		52		72	P3			
	84		92		102		84	P4	
	86		94		110		86	32	P5
	72		80		102		74	84	
	112		120		120		112	112	P6
									P7

Figura A.47: Tabla durante la segunda etapa de agregaciones.

P0									
0	28	P1							
	54		54	P2					
1	44		52		72	P3			
	84		92		102		84	P4	
	86		94		110		86	32	P5
	72		80		102		74	84	
	112		120		120		112	112	P6
									P7

Figura A.48: Tabla durante la segunda etapa de agregaciones.

P0	0								
0	P1	28							
-28		0	P2	26					
0		20		-2	P3	44			
-39		-19		-31		5	P4	45	
-39		-19		-37		5	0	P5	47
-33		-13		-37		9	0		P6
-47		-27		-29		-3	-2	0	P7
									65

Figura A.49: Tabla durante la segunda etapa de agregaciones.

P0															
0-8	28	P1													
	54	1	54	P2											
1-8	44	8	52		72	P3									
	84	32	102		84	P4									
	86	34	110		86	0	32	P5							
	72	80	102	74	1	84	0	86	P6						
	112	120	120	112	112	112	1	112		112	P7				

Figura A.50: Tabla durante la segunda etapa de agregaciones.

P0													
	28	P1											
	54	1	54	P2									
1	44	0	52		72	P3							
	84		92		102		84	P4					
	86		94		110		86	0	32	P5			
	72		80		102		74	1	84	0	86	P6	
	112		120		120		112		112	1	112		P7

Figura A.51: Tabla durante la segunda etapa de agregaciones.

P0	0												
	-20	P1	8										
	-8		0	P2	46								
	0		0		18	P3	44						
	-39		-39		-11		5	P4	45				
	-39		-39		-17		5		0	P5	47		
	-33		-33		-17		9		0		0	P6	39
	-47		-47		-9		-3		-2		0		-8
												P7	65

Figura A.52: Tabla durante la segunda etapa de agregaciones.

P0												
	28	P1										
	54	1-8	54	P2								
1-25	44	0-8	52	8	72	P3						
	84		92	102		84	P4					
	86		94	110		86	0	32	P5			
	72		80	102		74	1	84	0	86	P6	
	112		120	120		112	112	1	112		112	P7

Figura A.53: Tabla durante la segunda etapa de agregaciones.

P0												
	28	P1										
	54	1/2	54	P2								
	44	1/2	52	1/2	72	P3						
	84		92	102		84	P4					
	86		94	110		86	0	32	P5			
	72		80	102		74	1	84	0	86	P6	
	112		120	120		112	112	1	112		112	P7

Figura A.54: Tabla durante la segunda etapa de agregaciones.

P0	0							
-11	P1	17						
-17	0	P2	37					
-9	0	0	P3	35				
-39	-30	-20	-4	P4	45			
-39	-30	-26	-4	0	P5	47		
-33	-24	-26	0	0	0	P6	39	
-47	-38	-18	-12	-2	0	-8	P7	65

Figura A.55: Tabla durante la segunda etapa de agregaciones.

A0								
1	28	A1						
1	54	54	A2					
1-8	44	52	72	A3				
	84	92	102	84	A4			
	86	94	110	86	0	32	A5	
0-8	72	80	102	74	1	84	86	A6
	112	120	120	112	112	112	112	A7

Figura A.56: Tabla durante la segunda etapa de agregaciones.

A0								
1	28	A1						
1	54	54	A2					
1	44	52	72	A3				
	84	92	102	84	A4			
	86	94	110	86	0	32	A5	
	72	80	102	74	1	84	86	A6
	112	120	120	112	112	112	112	A7

Figura A.57: Tabla durante la segunda etapa de agregaciones.

A0	0							
0	A1	28						
0	28	A2	54					
0	20	26	A3	44				
-30	-10	6	14	A4	54			
-48	-28	-18	-4	0	A5	38		
-42	-22	-18	0	0	-18	A6	30	
-38	-18	8	6	16	0	-8	A7	74

Figura A.58: Tabla durante la segunda etapa de agregaciones.

A0								
1-8	28	A1						
1-8	54	54	A2					
1	44	52	72	A3				
	84	92	102	84	A4			
	86	94	110	86	0	32	A5	
	72	80	102	74	1	84	86	A6
	112	120	120	112	112	112	112	A7

Figura A.59: Tabla durante la segunda etapa de agregaciones.

[illegible]

Figura A.60: Tabla durante la segunda etapa de agregaciones.

A0	0									
0	A1	28								
-28	0	A2	26							
0	20	-2	A3	44						
-30	-10	-22	14	A4	54					
-48	-28	-46	-4	0	A5	38				
-42	-22	-46	0	0	-18					
-38	-18	-20	6	16	0	A6	30			
						-8	A7	74		

Figura A.61: Tabla durante la segunda etapa de agregaciones.

A0									
0-5	28								
	54	A1							
6-9		54	A2						
	44	B	72	A3					
	84	92	102	84					
	86	34	110	86	C	32	A4		
	72	80	102	74	D	84		E	
	112	120	120	112	112	F	116		G

Figura A.62: Tabla durante la segunda etapa de agregaciones.

[illegible]

Figura A.63: Tabla durante la segunda etapa de agregaciones.

A0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
----	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Figura A.64: Tabla durante la segunda etapa de agregaciones.

A0									
28	A1								
54	1-8	54	A2						
1-28	0-8	52	8	72	A3				
84		92	102	84	A4				
86	94	110	86	0	92	A5			
72	80	102	0	74	1	84	86	A6	
112	120	120	112	112	1	112	112	A7	

Figura A.65: Tabla durante la segunda etapa de agregaciones.

A0									
28	A1								
54	1/2	54	A2						
44	1/2	52	1/2	72	A3				
84		92	102	84	A4				
86	94	110	86	0	92	A5			
72	80	102	0	74	1	84	86	A6	
112	120	120	112	112	1	112	112	A7	

Figura A.66: Tabla durante la segunda etapa de agregaciones.

A0	0								
-11	A1	17							
-17	0	A2	37						
-9	0	0	A3	35					
-39	-30	-20	-4	A4	45				
-39	-30	-26	-4	0	A5	47			
-33	-24	-26	0	0	0	A6	39		
-47	-38	-18	-12	-2	0	-8	A7	65	

Figura A.67: Tabla durante la segunda etapa de agregaciones.

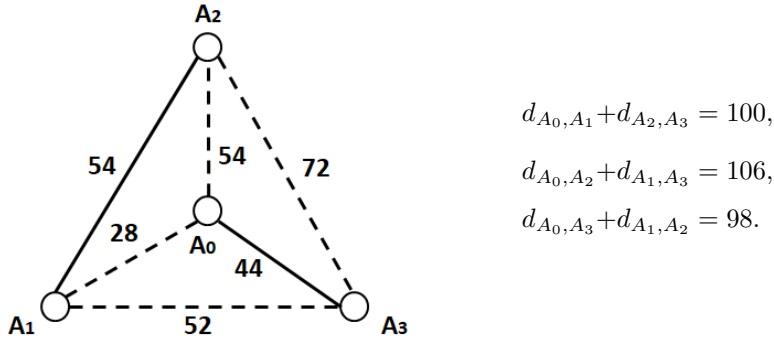


Figura A.68: Diagrama con las relaciones.

grupo de agregados A_1 , A_2 y A_3 , además vemos como cualquiera dos de ellos puede agregarse uno con otro, dejando que el tercero se agregue con el terminal A_0 . Dantzig y Ramser afirman que por prueba y error se puede decidir cuál de las tres alternativas corresponde a una distancia total más pequeña y utilizan el diagrama que se muestra en la Figura A.68 para realizar el cálculo.

En este diagrama A_1 , A_2 y A_3 se colocan en los vértices de un triángulo y A_0 se coloca en el centro, donde se conecta con el resto de los vértices. Las distancias correspondientes a las 6 posibles agregaciones se toman de la Figura A.66 y se colocan en las líneas correspondientes de la Figura A.68. Se ve fácilmente que las agregaciones que producen la distancia más pequeña son $(A_1A_2) + (A_0A_3)$, ya que $d_{A_0,A_3} + d_{A_1,A_2} = 44 + 54 = 98$, mientras que otras posibilidades que existen son $(A_0A_1) + (A_2A_3)$ y $(A_0A_2) + (A_1A_3)$ cuyas distancias suman 100 y 106 respectivamente.

Una vez relacionadas esas agregaciones hay que intentar relacionar las demás otra vez realizando el mismo procedimiento que hemos estado haciendo hasta ahora partiendo desde la tabla que se ve en la Figura A.69. El siguiente paso es calcular la función Delta como hemos estado haciendo hasta ahora y seleccionar el máximo valor $\delta_{i,j}$ que corresponde a $\delta_{4,7}$ (ver Figura A.70). A partir de ahí aparece la variable θ y se ajustan los $x_{i,j}$ como hemos hecho anteriormente (ver Figura A.71). Posteriormente como se ve en la Figura A.72 se le da el valor $\theta = \frac{1}{2}$ a la variable y eliminamos de la tabla a la entrada (A_0, A_3) en vez de la entrada (A_4, A_6) a pesar de que $d_{A_0,A_3} < d_{A_4,A_6}$ ya que si cogemos la segunda entrada si seguimos haciendo cuentas y queremos reajustar los $x_{i,j}$ en todos los casos llegamos a alguna contradicción. Por lo que sacando del conjunto básico la entrada (A_0, A_3) y recalculando los π_i y los $\delta_{i,j}$ llegamos a la tabla de la Figura A.73, donde vemos que ya no hay más $\delta_{i,j} > 0$. Así que la solución final es la que se obtiene de la Figura A.72, pero en este caso vuelven a aparecer valores fraccionarios en la solución y se solucionan con la técnica de los triángulos utilizada anteriormente como se puede ver en las Figura A.74 y A.75.

Finalmente, después de todas las cuentas hemos llegado a la siguiente

A0							
1	44	A3					
84		84	A4				
86		86	0	32	A5		
72	0	74	1	84		86	A6
112	112		112	1	112		A7

Figura A.69: Tabla durante la segunda etapa de agregaciones.

A0	0						
0	A3	44					
-30	14		A4	54			
-48	-4	0		A5	38		
-42	0	0		-18	A6	30	
-38	6	16	0		-8	A7	74

Figura A.70: Tabla durante la segunda etapa de agregaciones.

solución: (A_1A_2) , (A_5A_7) , (A_4A_6) , A_3 , que si sustituimos por las agregaciones obtenidas en la primera etapa tenemos la solución $(P_1P_2P_3P_4)$, $(P_7P_{12}P_{11}P_9)$, $(P_6P_{10}P_8)$, P_5 .

A0							
1-28	44	A3					
84		84	A4				
86		86	0+8	32	A5		
72	0+28	74	1-28	84		86	A6
112		112	5	112	1-8	112	A7

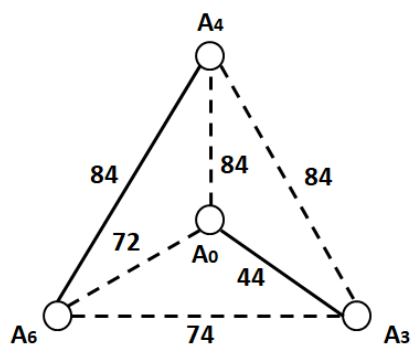
Figura A.71: Tabla durante la segunda etapa de agregaciones.

A0							
44	A3						
84		84	A4				
86		86	1/2	32	A5		
72	1	74	0	84		86	A6
112		112	1/2	112	1/2	112	A7

Figura A.72: Tabla durante la segunda etapa de agregaciones.

A0	0						
-8	A3	36					
-38	-2		A4	46			
-40	-4	0		A5	46		
-34	0	0		-2	A6	38	
-46	-10	0	0		-8	A7	66

Figura A.73: Tabla durante la segunda etapa de agregaciones.

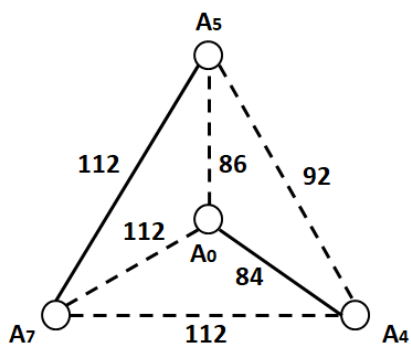


$$d_{A_0, A_3} + d_{A_4, A_6} = 128,$$

$$d_{A_0, A_4} + d_{A_3, A_6} = 158,$$

$$d_{A_0, A_6} + d_{A_3, A_4} = 156.$$

Figura A.75: Diagrama con las relaciones.



$$d_{A_0, A_4} + d_{A_5, A_7} = 196,$$

$$d_{A_0, A_5} + d_{A_4, A_7} = 198,$$

$$d_{A_0, A_7} + d_{A_4, A_5} = 204.$$

Figura A.74: Diagrama con las relaciones.

Apéndice B

Ejemplo numérico de Clarke y Wright

Primera iteración

Para empezar, suponemos la solución que se puede ver en la Figura B.1, donde se elige como primera solución aquella con $x_{y,0} = 2$ donde $y \in \{1, 2, \dots, 12\}$, es decir que todos los puntos estacionarios están relacionados con el terminal P_0 . En esta tabla también vemos la primera parte del algoritmo donde las casillas que no pueden ser seleccionadas (hay una serie de condiciones que tienen que cumplir las cuales están explicadas en el capítulo 2) están de color rojo mientras que la casilla que tiene mayor ahorro (valor de abajo a la izquierda cuyo cálculo está explicado en el capítulo 2) está en color verde, siendo la siguiente entrada que va a entrar en la solución en la siguiente tabla. Además empezamos asignando todos los vehículos de menos de 4000, ya que de momento todos los viajes requieren una carga menor a esa cantidad.

Segunda iteración

Como el máximo ahorro estaba en la celda (11 : 12), se cambia el valor de $x_{11,12} = 1$, reajustando los valores de las celdas (0 : 11) y (0 : 12), al igual que se reajustan los valores de las cargas de la siguiente forma:

$$Q_{11}^2 = Q_{12}^2 = Q_{11}^1 + Q_{12}^1 = 1700 + 1100 = 2800.$$

Además en la segunda tabla de la Figura B.2 es necesario un vehículo menos de capacidad menor a 4000. Finalmente, la distancia mínima se disminuye el valor correspondiente al ahorro anteriormente seleccionado, es decir $728 - 92 = 636$, y toca elegir el siguiente máximo ahorro, el cual corresponde es 84, perteneciente a las celdas (10 : 11) y (10 : 12) de las cuales de forma aleatoria se selecciona la segunda y posteriormente veremos que da igual seleccionar una o la otra.

Tercera iteración

Al igual que en la anterior iteración se cambia a uno el valor de la celda correspondiente al anterior máximo ahorro, el cual estaba en la celda (10 : 12),

Figura B.1: Tablas al principio del algoritmo.

Figura B.1: Tablas al principio del algoritmo.

Figura B.2: Tablas en la segunda iteración.

Figura B.2: Tablas en la segunda iteración.

Q	P0																								
1200	2 9	P1																							
1700	2 14	18	5	P2																					
1500	2 21	18	12	28	7	P3																			
1400	2 23	10	22	20	17	34	10	P4																	
1700	2 22	10	21	20	16	22	21	26	19	P5															
1400	2 25	10	24	16	23	16	30	20	28	38	9	P6													
1200	2 32	10	31	20	26	26	27	30	25	44	10	50	7	P7											
1900	2 36	10	35	20	30	20	37	24	35	42	16	50	11	58	10	P8									
1800	3 38	10	37	16	36	16	43	20	41	38	22	50	13	54	16	68	6	P9							
4400	1 42	10	41	20	36	32	31	36	29	44	20	50	17	64	10	72	6	68	12	P10					
4400	1 50	10	19	20	44	34	37	42	31	44	28	50	25	64	18	72	14	76	12	84	8	P11			
-		52	10	51	20	46	34	39	46	29	44	30	50	27	64	20	72	16	70	20	84	10	92	10	P12

VEHÍCULO	≤ 4000	> 4000	> 5000	> 6000
DISPONIBLES	∞	7	4	0
ASIGNADOS	9	1	0	0

DISTANCIA MINIMA552

Figura B.3: Tablas en la tercera iteración.

y se reajustan los valores $x_{0,10}$ y $x_{0,12}$ disminuyendo en una unidad. Esto hace que $x_{0,12} = 0$, por lo que por la condición 1 todas las celdas de la forma $(y : 12)$ ya no van a poder ser seleccionadas y por eso se ponen de color rojo en la primera tabla de la Figura B.3. Sin embargo no son las únicas celdas que están en rojo además de las ya mencionadas anteriormente, ya que la celda $(10 : 11)$, correspondiente a la otra opción de selección en la anterior iteración, no puede ser seleccionada ya que incumpliría la condición 2. En cuanto a las cargas, Q_{12} toma el valor cero debido que $x_{0,12} = 0$, mientras que Q_{10} y Q_{11} se reajustan de la siguiente forma:

$$Q_{10}^3 = Q_{11}^3 = Q_{10}^2 + Q_{11}^2 = 1600 + 2800 = 4400.$$

Como la carga combinada de visitar P_{10} , P_{11} y P_{12} es igual a 4400, esta ruta ya no puede ser combinada con ningún punto cuya carga exceda de 1600, porque no disponemos de ningún vehículo con carga superior a 6000. Esto provoca que debamos modificar la segunda tabla de la Figura B.3, correspondiente a los vehículos, y que una serie de celdas no cumplan la condición 3 debido a que los valores de sus respectivas cargas hacen que la suma de ambas sea superior a 6000 como acabamos de indicar. Así que esas celdas al igual que las anteriores se muestran de color rojo en la primera tabla de la Figura 2.8. Finalmente se calcula el valor de la distancia mínima de forma similar, siendo en este caso igual a 552, y se selecciona el nuevo máximo ahorro de las celdas seleccionables, que corresponde al valor 68 de la celda (8 : 9).

Cuarta iteración

Al igual que en la anterior iteración se cambia a uno el valor de la celda correspondiente al anterior máximo ahorro, el cual estaba en la celda (8 : 9), y se reajustan los valores $x_{0,8}$ y $x_{0,9}$ disminuyendo en una unidad. En cuanto a las cargas, Q_8 y Q_9 se reajustan de la siguiente forma:

$$Q_8^4 = Q_9^4 = Q_8^3 + Q_9^3 = 1900 + 1800 = 3700.$$

Figura B.4: Tablas en la cuarta iteración.

Quinta iteración

$$Q_7^5 = Q_{10}^5 = Q_7^4 + Q_{10}^4 = 1200 + 4400 = 5600.$$

Esta carga de 5600 a parte de modificar la segunda tabla de la Figura B.5, correspondiente a los vehículos ya que provoca que haya que usar otro vehículo de más de 5000, también hace que una serie de celdas no cumplan la condición 3 debido a que los valores de sus respectivas cargas hacen que la suma de ambas sea superior a 6000 siendo este el valor más grande que los vehículos pueden llevar. Así que esas celdas al igual que las anteriores se muestran de color rojo en la primera tabla de la Figura B.5. Finalmente se calcula el valor de la distancia mínima de forma similar, siendo en este caso igual a 420, y se selecciona el nuevo máximo ahorro de las celdas seleccionables, que corresponde al valor 50 de la celda (6 : 8).

[illegible]

Figura B.6: Tablas en la sexta iteración.

Finalmente se calcula el valor de la distancia mínima de forma similar, siendo en este caso igual a 336, y se selecciona el nuevo máximo ahorro de las celdas seleccionables, que corresponde al valor 28 de la celda (2 : 3).

Octava iteración

Al igual que en la anterior iteración se cambia a uno el valor de la celda correspondiente al anterior máximo ahorro, el cual estaba en la celda $(2 : 3)$, y se reajustan los valores $x_{0,2}$ y $x_{0,3}$ disminuyendo en una unidad. Esto hace que $x_{0,3} = 0$, por lo que por la condición 1 todas las celdas de la forma $(y : 3)$ ya no van a poder ser seleccionadas y por eso se ponen de color rojo en la primera tabla de la Figura B.8. Sin embargo no son las únicas celdas que están en rojo además de las ya mencionadas anteriormente, ya que las celdas de la forma $(y : z)$ con $y, z \in \{2, 4, 5\}$ no pueden ser seleccionadas ya que incumplirían la condición 2. En cuanto a las cargas, Q_3 toma el valor cero debido que $x_{0,3} = 0$, mientras que Q_2 y Q_4 se reajustan de la siguiente forma:

$$Q_2^8 = Q_4^8 = Q_2^7 + Q_4^7 = 1700 + 2900 = 4600.$$

Esta carga de 4600 a parte de modificar la segunda tabla de la Figura B.8, correspondiente a los vehículos, ya que provoca que haya que usar otro vehículo de más de 4000, también hace que una serie de celdas no cumplan la condición 3, debido a que los valores de sus respectivas cargas hacen que la suma de ambas sea superior a 6000 siendo este el valor más grande que los vehículos pueden llevar. Así que esas celdas al igual que las anteriores se muestran de color rojo en la primera tabla de la Figura B.8. Finalmente se calcula el valor de la distancia mínima de forma similar, siendo en este caso igual a 308, y se selecciona el nuevo máximo ahorro de las celdas seleccionables, que corresponde al valor 18 de la celda (1 : 2).

Apéndice C

Problema de emparejamiento de peso máximo

A continuación voy a mostrar un texto realizado por Joris VR [16] donde se explica en qué consiste este problema, primero con una pequeña introducción y posteriormente con su implementación en Python.

C.1. Emparejamientos en grafos

En la teoría de grafos, un emparejamiento es un subconjunto de aristas tal que ninguno de las aristas seleccionadas comparte un vértice común.

Un emparejamiento máximo de cardinalidad (“maximum cardinality matching”) es un emparejamiento que contiene el mayor número posible de aristas o, de manera equivalente, el mayor número posible de vértices. Además un emparejamiento perfecto (“perfect matching”) es un emparejamiento que contiene todos los vértices. Con respecto a un grafo ponderado, un emparejamiento de peso máximo (“maximum weighted matching”) es un emparejamiento para el cual la suma de los pesos de las aristas coincidentes es lo más grande posible. En la Figura C.1 se ve un ejemplo donde se ha encontrado un emparejamiento con valor de 15 en el cual están emparejados los vértices de color rojo y azul.

C.2. Algoritmos

Encontrar un emparejamiento de peso máximo está lejos de ser trivial. Por ejemplo el emparejamiento de peso máximo de peso en la Figura C.1 contiene solo 4 de 6 vértices y aunque existen emparejamientos perfectos para este grafo, ninguno de ellos logra el peso máximo. Además también se puede ver que el borde con el peso más grande (el borde entre c y d con peso 9) no es parte del emparejamiento con peso máximo.

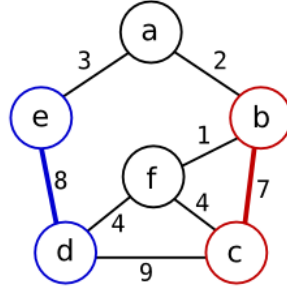


Figura C.1: Ejemplo de emparejamientos.

El primer algoritmo con complejidad polinomial para los emparejamientos máximos fue encontrado por Edmonds [13] y posteriormente fue mejorado por Gabow [9] y Z. Galil [31]. Sin embargo, Joris VR [16] da un código Phyton que implementa un algoritmo con complejidad de $O(n^3)$ descrito por Galil [31], el cual calcula un emparejamiento de peso máximo en un gráfico general (no necesariamente bipartito). Además, alternativamente, se calcula un emparejamiento de cardinalidad máxima con el mayor peso posible entre todos los emparejamientos de cardinalidad máxima.